

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5373385号  
(P5373385)

(45) 発行日 平成25年12月18日(2013.12.18)

(24) 登録日 平成25年9月27日(2013.9.27)

(51) Int.Cl. F I  
**G05B 19/05 (2006.01)** G O 5 B 19/05 L  
**H04L 12/28 (2006.01)** H O 4 L 12/28 2 O 3

請求項の数 24 外国語出願 (全 64 頁)

(21) 出願番号	特願2008-327645 (P2008-327645)	(73) 特許権者	512132022
(22) 出願日	平成20年12月24日(2008.12.24)		フィッシャーローズマウント システム
(65) 公開番号	特開2009-217812 (P2009-217812A)		ズ、インコーポレイテッド
(43) 公開日	平成21年9月24日(2009.9.24)		アメリカ合衆国 テキサス州 78681
審査請求日	平成23年12月20日(2011.12.20)		ラウンド ロック ウェスト ルイス
(31) 優先権主張番号	11/968,087		ヘナ ブルバード 1100 ビルディン
(32) 優先日	平成19年12月31日(2007.12.31)		グ 1
(33) 優先権主張国	米国 (US)	(74) 代理人	100079049
			弁理士 中島 淳
		(74) 代理人	100084995
			弁理士 加藤 和詳
		(74) 代理人	100085279
			弁理士 西元 勝一

最終頁に続く

(54) 【発明の名称】 フィールドバスネットワーク内で動作する制御モジュールのための同期ブロック実行の自動構成

(57) 【特許請求の範囲】

【請求項1】

制御モジュールの機能ブロックを、プロセス制御ネットワーク内の装置内で実行されるように自動的に割り当てるための、プロセッサ上で動作可能なシステムであって、前記プロセス制御ネットワークは、1つ以上のフィールド装置にデータバスを介して接続された入出力装置に通信可能に結合されたプロセスコントローラを有し、

前記システムは、コンピュータ読み取り可能メモリと、前記プロセッサ上で実行するための、前記コンピュータ読み取り可能メモリ上に記憶されたルーチンとを具備し、前記ルーチンは、前記プロセッサ上で実行される場合、

前記制御モジュールのフォワードフロー経路上に配置された機能ブロックの組であって、入力機能ブロックと出力機能ブロックと1つ以上の中間機能ブロックとを含む、機能ブロックの組を決定し、

前記制御モジュールの前記フォワードフロー経路上の、前記入力機能ブロックと前記出力機能ブロックとのそれぞれが、前記データバス上の前記1つ以上のフィールド装置内で実行されるように割り当てられるかどうかを判定し、及び、

前記制御モジュールの前記フォワードフロー経路上の、前記入力機能ブロックと前記出力機能ブロックとの両方が、前記データバス上の前記1つ以上のフィールド装置内で実行されるように割り当てられるならば、前記制御モジュールの前記フォワードフロー経路上の、前記1つ以上の中間機能ブロックを、前記入出力装置内で実行されるように割り当て、前記制御モジュールの前記フォワードフロー経路上の前記入力機能ブロックと前記出力

10

20

機能ブロックとの両方が、前記データベース上の前記1つ以上のフィールド装置内で実行するように割り当てられていないならば、前記フォワードフロー経路上の、前記1つ以上の中間機能ブロックを前記コントローラ内で実行するように割り当てる、システム。

【請求項2】

前記ルーチンは、前記プロセッサ上で実行される場合、前記制御モジュールの前記フォワードフロー経路上の、前記入力機能ブロックと前記出力機能ブロックとのうちの1つ以上が、前記データベース上の前記1つ以上のフィールド装置のうちのいずれかの中で実行されるように割り当てられないならば、前記制御モジュールの前記フォワードフロー経路上の、前記1つ以上の中間機能ブロックのそれぞれを、前記プロセスコントローラ内で実行されるように割り当てる、

10

請求項1に記載のシステム。

【請求項3】

事前に確立される通信スケジュールに従って、前記データベースを介して通信が行われ、及び、

前記ルーチンは、前記プロセッサ上で実行される場合、さらに、前記入力機能ブロックと前記出力機能ブロックと前記1つ以上の中間機能ブロックとのそれぞれを、前記事前に確立される通信スケジュール内で同期して動作するようにスケジュールする、

請求項2に記載のシステム。

【請求項4】

20

前記事前に確立される通信スケジュールは、繰り返されるマクロサイクルを含み、及び、

前記ルーチンは、前記プロセッサ上で実行される場合、前記入力機能ブロックと前記出力機能ブロックと前記1つ以上の中間機能ブロックとのそれぞれが、各マクロサイクルの間に少なくとも1回、前記データベースを介して通信するようにスケジュールされる、前記データベースのためのマクロサイクルを作成することによって、前記入力機能ブロックと前記出力機能ブロックと前記1つ以上の中間機能ブロックとのそれぞれを、前記事前に確立される通信スケジュール内でスケジュールする、

請求項3に記載のシステム。

【請求項5】

30

前記ルーチンは、前記プロセッサ上で実行される場合、前記入力機能ブロックと前記出力機能ブロックと前記1つ以上の中間機能ブロックとのうちの1つが、前記入力機能ブロックと前記出力機能ブロックと前記1つ以上の中間機能ブロックとのうちの別の1つよりも速いレートで実行されるようにスケジュールする、

請求項4に記載のシステム。

【請求項6】

前記ルーチンは、前記プロセッサ上で実行される場合、さらに、前記事前に確立される通信スケジュールのための前記マクロサイクルを作成し、前記マクロサイクルの中では、前記入力機能ブロックと前記出力機能ブロックと前記1つ以上の中間機能ブロックとのうちの前記1つが、前記入力機能ブロックと前記出力機能ブロックと前記1つ以上の中間機能ブロックとのうちの前記別の1つよりも多くの回数、前記マクロサイクルの間に動作するようにスケジュールされる、

40

請求項5に記載のシステム。

【請求項7】

マルチプライヤを指定する、前記制御モジュールの前記機能ブロックのうちの少なくとも1つのためのインジケータをさらに含み、前記マルチプライヤは、前記制御モジュールの前記機能ブロックのうちの前記少なくとも1つについての、相対的な実行レートを指示し、及び、

前記ルーチンは、前記プロセッサ上で実行される場合、前記インジケータを使用して、前記入力機能ブロックと前記出力機能ブロックと前記1つ以上の中間機能ブロックとのう

50

ちの前記1つが、前記入力機能ブロックと前記出力機能ブロックと前記1つ以上の中間機能ブロックとのうちの前記別の1つよりも速いレートで実行されるようにスケジュールする、

請求項6に記載のシステム。

【請求項8】

前記制御モジュールの前記機能ブロックが前記入出力装置に割り当てられることが可能かどうかを示す、前記制御モジュールに関連付けられたユーザ設定可能なプロパティをさらに含み、及び、

前記ルーチンは、前記プロセッサ上で実行される場合、前記制御モジュールの前記ユーザ設定可能なプロパティが、前記制御モジュールの機能ブロックは前記入出力装置に割り当てられるべきではないということを指示するならば、たとえ、前記制御モジュールの前記フォワードフロー経路上の前記入力機能ブロックと前記出力機能ブロックとの両方が、前記データベース上の前記1つ以上のフィールド装置内で実行されるように割り当てられるとしても、前記制御モジュールの前記フォワードフロー経路上の前記1つ以上の中間機能ブロックを、前記プロセスコントローラ内で実行されるように割り当てる、

請求項7に記載のシステム。

【請求項9】

任意の制御モジュールの機能ブロックが前記入出力装置に割り当てられることが可能かどうかを示す、前記入出力装置に関連付けられたユーザ設定可能なプロパティをさらに含み、及び、

前記ルーチンは、前記プロセッサ上で実行される場合、前記入出力装置の前記ユーザ設定可能なプロパティが、機能ブロックは前記入出力装置に割り当てられるべきではないということを指示するならば、たとえ、前記制御モジュールの前記フォワードフロー経路上の前記入力機能ブロックと前記出力機能ブロックとの両方が、前記データベース上の前記1つ以上のフィールド装置内で実行されるように割り当てられるとしても、前記制御モジュールの前記フォワードフロー経路上の前記1つ以上の中間機能ブロックを、前記プロセスコントローラ内で実行されるように割り当てる、

請求項7に記載のシステム。

【請求項10】

前記ルーチンは、前記プロセッサ上で実行される場合、前記入出力装置が、前記制御モジュールの前記フォワードフロー経路上の前記1つ以上の中間機能ブロックのそれぞれをサポートするかどうかを調べて判定し、

前記入出力装置が、前記1つ以上の中間機能ブロックのうちのいずれか1つでもサポートしないならば、たとえ、前記制御モジュールの前記フォワードフロー経路上の前記入力機能ブロックと前記出力機能ブロックとの両方が、前記データベース上の前記1つ以上のフィールド装置内で実行されるように割り当てられるとしても、前記制御モジュールの前記フォワードフロー経路上の前記1つ以上の中間機能ブロックのそれぞれを、前記プロセスコントローラ内で実行されるように割り当てる、

請求項7に記載のシステム。

【請求項11】

前記ルーチンは、前記プロセッサ上で実行される場合、前記入出力装置が、前記制御モジュールの前記フォワードフロー経路上の前記1つ以上の中間機能ブロックのうちのすべてを実行するための十分な容量を有するかどうかを調べて判定し、

前記入出力装置が、前記1つ以上の中間機能ブロックのうちのすべてを実行するための十分な容量を有さないならば、たとえ、前記制御モジュールの前記フォワードフロー経路上の前記入力機能ブロックと前記出力機能ブロックとの両方が、前記データベース上の前記1つ以上のフィールド装置内で実行されるように割り当てられるとしても、前記制御モジュールの前記フォワードフロー経路上の前記1つ以上の中間機能ブロックを、前記プロセスコントローラ内で実行されるように割り当てる、

請求項7に記載のシステム。

10

20

30

40

50

## 【請求項 1 2】

前記ルーチンは、前記プロセッサ上で実行される場合、前記ルーチンが、前記制御モジュールの前記フォワードフロー経路上の前記 1 つ以上の中間機能ブロックを、前記入出力装置内で実行されるように割り当てないならば、ユーザに通知する、

請求項 1 1 に記載のシステム。

## 【請求項 1 3】

制御モジュールの機能ブロックを、プロセス制御ネットワーク内での実行のために装置に割り当てる方法であって、前記プロセス制御ネットワークは、1 つ以上のフィールド装置にデータベースを介して接続された入出力装置に通信可能に結合されたプロセスコントローラを有し、前記方法は、

前記制御モジュールのフォワードフロー経路上に配置された機能ブロックの組を決定することであって、機能ブロックの前記組は、入力機能ブロックと、出力機能ブロックと、1 つ以上の中間機能ブロックとを含む、ことと、

前記制御モジュールの前記フォワードフロー経路上の、前記入力機能ブロックと前記出力機能ブロックとのそれぞれが、前記データベース上の前記 1 つ以上のフィールド装置内で実行されるように割り当てられるかどうかを判定することと、

前記制御モジュールの前記フォワードフロー経路上の、前記入力機能ブロックと前記出力機能ブロックとの両方が、前記データベース上の前記 1 つ以上のフィールド装置内で実行されるように割り当てられるならば、前記制御モジュールの前記フォワードフロー経路上の、前記 1 つ以上の中間機能ブロックのうち少なくとも 1 つを、前記入出力装置内で実行されるように割り当て、前記制御モジュールの前記フォワードフロー経路上の前記入力機能ブロックと前記出力機能ブロックとの両方が、前記データベース上の前記 1 つ以上のフィールド装置内で実行するように割り当てられていないならば、前記制御モジュールの前記フォワードフロー経路上の、前記 1 つ以上の中間機能ブロックの少なくとも 1 つを前記コントローラ内で実行するように割り当てることを含む、方法。

## 【請求項 1 4】

前記制御モジュールの前記フォワードフロー経路上の、前記入力機能ブロックと前記出力機能ブロックとのうちの 1 つ以上が、前記データベース上の前記 1 つ以上のフィールド装置内で実行されるように割り当てられないならば、前記制御モジュールの前記フォワードフロー経路上の、前記 1 つ以上の中間機能ブロックのそれぞれを、前記プロセスコントローラ内で実行されるように割り当ててをさらに含む、

請求項 1 3 に記載の、制御モジュールの機能ブロックを割り当てる方法。

## 【請求項 1 5】

前記制御モジュールの前記フォワードフロー経路上の、前記 1 つ以上の中間機能ブロックのうち少なくとも 1 つを、前記入出力装置内で実行されるように割り当ててことは、前記制御モジュールの前記フォワードフロー経路上の、前記 1 つ以上の中間機能ブロックのうちそれぞれを、前記入出力装置内で実行されるように割り当ててことを含む、請求項 1 3 に記載の、制御モジュールの機能ブロックを割り当てる方法。

## 【請求項 1 6】

事前に確立される通信スケジュールに従って、前記データベースを介して通信が行われ、及び、

前記入力機能ブロックと前記出力機能ブロックと前記 1 つ以上の中間機能ブロックのうちの前記少なくとも 1 つとのそれぞれを、前記事前に確立される通信スケジュール内で同期して動作するようにスケジュールすることをさらに含む、

請求項 1 5 に記載の、制御モジュールの機能ブロックを割り当てる方法。

## 【請求項 1 7】

前記入力機能ブロックと前記出力機能ブロックと前記 1 つ以上の中間機能ブロックのうちの前記少なくとも 1 つとのそれぞれをスケジュールすることは、前記入力機能ブロックと前記出力機能ブロックと前記 1 つ以上の中間機能ブロックとのうちの 1 つが、前記入力機能ブロックと前記出力機能ブロックと前記 1 つ以上の中間機能ブロックとのうちの別の

10

20

30

40

50

1 つよりも速いレートで実行されるようにスケジュールすることを含む、  
請求項 16 に記載の、制御モジュールの機能ブロックを割り当てる方法。

【請求項 18】

前記事前に確立される通信スケジュールのためのマクロサイクルを作成することであって、前記マクロサイクルの中では、前記入力機能ブロックと前記出力機能ブロックと前記 1 つ以上の中間機能ブロックとのうちの前記 1 つが、前記入力機能ブロックと前記出力機能ブロックと前記 1 つ以上の中間機能ブロックとのうちの前記別の 1 つよりも多くの回数、前記マクロサイクルの間に動作するようにスケジュールされる、ことをさらに含む、  
請求項 17 に記載の、制御モジュールの機能ブロックを割り当てる方法。

【請求項 19】

前記入力機能ブロックと前記出力機能ブロックと前記 1 つ以上の中間機能ブロックとのうちの 1 つ以上のための、マルチプライヤをユーザが指定することを可能にすることであって、前記マルチプライヤは、関連する機能ブロックについての相対的な実行レートを指示する、ことをさらに含む、

請求項 18 に記載の、制御モジュールの機能ブロックを割り当てる方法。

【請求項 20】

前記入出力装置が、前記制御モジュールの前記フォワードフロー経路上の前記 1 つ以上の中間機能ブロックのうちの前記少なくとも 1 つを実行するための十分な容量を有するかどうかを調べて判定することと、

前記入出力装置が、前記 1 つ以上の中間機能ブロックのうちの前記少なくとも 1 つを実行するための十分な容量を有さないならば、たとえ、前記制御モジュールの前記フォワードフロー経路上の前記入力機能ブロックと前記出力機能ブロックとの両方が、前記データベース上の前記 1 つ以上のフィールド装置内で実行されるように割り当てられるとしても、前記制御モジュールの前記フォワードフロー経路上の前記 1 つ以上の中間機能ブロックのうちの前記少なくとも 1 つを、前記プロセスコントローラ内で実行されるように割り当てることをさらに含む、

請求項 13 に記載の、制御モジュールの機能ブロックを割り当てる方法。

【請求項 21】

ユーザインタフェース装置と、  
第 1 の通信ネットワークを介して前記ユーザインタフェース装置に通信可能に結合された、プロセスコントローラと、

データベースと、前記データベースに通信可能に接続された 1 つ以上のフィールドバスフィールド装置とを含む、フィールドバスセグメントであって、前記 1 つ以上のフィールドバスフィールド装置は、事前に確立される通信スケジュールを使用して、前記データベースを介して通信する、フィールドバスセグメントと、

前記データベースに直接接続され、そして、前記プロセスコントローラと前記データベースとの間に通信可能に接続された、入出力装置と、

プロセス制御ループを実行するために一緒に動作する複数の通信可能にリンクされた機能ブロックを有する、制御モジュールであって、前記複数の通信可能にリンクされた機能ブロックは、入力機能ブロックと出力機能ブロックと少なくとも 1 つの中間機能ブロックとを含み、前記入力機能ブロックと前記出力機能ブロックとは前記 1 つ以上のフィールドバスフィールド装置内に配置され、前記少なくとも 1 つの中間機能ブロックは前記入出力装置内に配置され、及び、前記入力機能ブロックと前記出力機能ブロックと前記少なくとも 1 つの中間機能ブロックとのそれぞれは、前記事前に確立される通信スケジュールを使用して前記データベースを介して通信する、制御モジュールとを具備する、  
プロセス制御ネットワークであって、

プロセッサと、コンピュータ読み取り可能メモリと、前記コンピュータ読み取り可能メモリ内に記憶されたルーチンとをさらに含み、前記ルーチンは、前記制御モジュールの前記機能ブロックを、前記プロセス制御ネットワーク内の特定の装置内での実行のために、自動的に割り当てるために、前記プロセッサ上で実行され、

10

20

30

40

50

前記ルーチンは、

前記制御モジュールのフォワードフロー経路上に配置された機能ブロックの組であって、前記入力機能ブロックと前記出力機能ブロックと前記少なくとも1つの中間機能ブロックとを含む、機能ブロックの組を決定し、

前記制御モジュールの前記フォワードフロー経路上の、前記入力機能ブロックと前記出力機能ブロックとのそれぞれが、前記データバス上の前記1つ以上のフィールドバスフィールド装置内で実行されるように割り当てられるかどうかを判定し、及び、

前記制御モジュールの前記フォワードフロー経路上の、前記入力機能ブロックと前記出力機能ブロックとの両方が、前記データバス上の前記1つ以上のフィールドバスフィールド装置内で実行されるように割り当てられるならば、前記制御モジュールの前記フォワードフロー経路上の、前記少なくとも1つの中間機能ブロックを、前記入出力装置内で実行されるように割り当て、第2の制御モジュールのフォワードフロー経路上の入力機能ブロックと出力機能ブロックの1つ以上が、前記データバス上の前記1つ以上のフィールド装置内で実行するように割り当てられていないならば、前記第2の制御モジュールの前記フォワードフロー経路上の、少なくとも1つの中間機能ブロックを前記プロセスコントローラ内で実行するように割り当てる、

プロセス制御ネットワーク。

#### 【請求項22】

前記ルーチンは、前記プロセッサ上で実行される場合、前記入力機能ブロックと前記出力機能ブロックと前記少なくとも1つの中間機能ブロックとのうちの1つが、前記入力機能ブロックと前記出力機能ブロックと前記少なくとも1つの中間機能ブロックとのうちの別の1つよりも速いレートで、前記データバス上で実行されるようにスケジュールする、  
請求項21に記載のプロセス制御ネットワーク。

#### 【請求項23】

前記制御モジュールの前記機能ブロックが前記入出力装置に割り当てられることが可能かどうかを示す、前記制御モジュールに関連付けられたユーザ設定可能なプロパティをさらに含む、

請求項22に記載のプロセス制御ネットワーク。

#### 【請求項24】

任意の制御モジュールの機能ブロックが前記入出力装置に割り当てられることが可能かどうかを示す、前記入出力装置に関連付けられたユーザ設定可能なプロパティをさらに含む、

請求項22に記載のプロセス制御ネットワーク。

#### 【発明の詳細な説明】

#### 【技術分野】

#### 【0001】

本出願は、2006年9月29日に出願された、「プロセス制御システムにおける機能を実行するためのスケジュールを生成する方法および装置 (Methods and Apparatus to Generate Schedules to Execute Functions in a Process Control System)」と題された、米国特許出願第11/537,303号の一部継続出願であり、かつ、当該出願の優先権を主張するものである。当該出願の全開示内容は、参照により本明細書に明示的に援用される。

#### 【0002】

本発明は、一般に、プロセス制御システムに関し、より詳細には、制御モジュールのさまざまな機能ブロックをフィールドバス (Fieldbus) セグメント上の入出力カードに割り当てるシステムであって、制御モジュールの機能ブロックをフィールドバスセグメント内で同期式に実行するためのスケジュールを生成する、システムに関する。

#### 【背景技術】

#### 【0003】

10

20

30

40

50

化学プロセス、石油プロセス、またはその他のプロセスにおいて使用されるもののような、プロセス制御システムは、一般に、アナログバス、デジタルバス、またはアナログ/デジタルを組み合わせたバスを介して、少なくとも1つのホストまたはオペレータワークステーションに、および、1つ以上のフィールド装置に通信可能に結合された、1つ以上の集中型プロセスコントローラを含む。バルブ、バルブポジショナ、スイッチ、トランスミッタ（例えば、温度センサ、圧力センサ、および流量センサ）などであってもよい、フィールド装置は、バルブの開閉、およびプロセスパラメータの測定などの、プロセス内の機能を実行する。プロセスコントローラは、フィールド装置によって作成されたプロセス測定値を表す信号、および/またはフィールド装置に関連するその他の情報を表す信号を受信し、この情報を使用して制御ルーチンまたは制御モジュールを実施し、そして、次に、プロセスの動作を制御するためにバスまたはその他の通信線を介してフィールド装置に送信される、制御信号を生成する。プロセスの現在の状態を表示すること、プロセスの動作を変更することなどの、プロセスに関する所望の機能をオペレータが実行することを可能にするために、フィールド装置およびコントローラからの情報は、オペレータワークステーションによって実行される1つ以上のアプリケーションで利用できるようにされる場合がある。

10

**【0004】**

プロセス制御システムアプリケーションは、一般に、プロセス制御システム内のさまざまな機能または動作を実行するために構成されることが可能な、プロセス制御ルーチンを含む。例えば、プロセス制御ルーチンは、バルブ、モータ、ボイラ、ヒータなどを制御するために使用される場合がある。プロセス制御ルーチンは、さらに、フィールド装置、プラントエリアなどを監視するため、および、プロセス制御システムに関連付けられた情報を収集するために使用される場合がある。プロセス制御ルーチンを実施するために使用されるフィールド装置は、通常、データバスを介して、相互に、およびプロセスコントローラに結合される。しかし、それらのデータバスには、フィールド装置およびプロセスコントローラが相互に情報を通信するために利用可能な帯域幅の制限をはじめとする、資源の制限がある。場合によっては、すべてのフィールド装置とコントローラとが、それらのそれぞれの機能を実行し、そしてそれらの間で情報を通信するための、十分な時間をデータバス上で得られることを確実にするために、コントローラとフィールド装置とが、いつ、それらのそれぞれの機能を実行し、そしてデータバス上で通信すべきかを示す、スケジュールが生成される。

20

30

**【0005】**

プロセス制御システムの技術分野において周知の、ファウンデーション（FOUNDATION（登録商標））フィールドバス（Fieldbus）（FF）通信プロトコルおよび仕様は、ファウンデーション（FOUNDATION（登録商標））フィールドバス（Fieldbus）（FF）機能ブロックとして一般に知られている機能ブロックが、制御ループを実施するために、バルブ、トランスミッタなどのさまざまなフィールド装置内に配置され、そして通信リンクまたはバスを介して相互に通信することを可能にする、全デジタルの通信プロトコルである。特定の制御ループまたは機能を実施する、相互接続された機能ブロックの組は、一般に、プロセス制御モジュールと呼ばれ、そして、そのようなプロセス制御モジュールは、フィールドバスセグメントとも呼ばれるフィールドバス通信バスに接続された、さまざまな異なる装置内に配置される、機能ブロックを有する場合がある。一般に知られているように、特定のフィールドバスセグメント上の通信は、一般にオペレータまたは構成エンジニアによって、フィールドバスセグメントのセットアップ時または構成時に、セグメントのために生成される、通信スケジュールによって決定される。一般的に言えば、特定のフィールドバスセグメントのための通信スケジュールは、一組の繰り返されるマクロサイクルのそれぞれの間、フィールドバスセグメントのバスを介して行われる、同期通信のそれぞれを定義する。

40

**【先行技術文献】****【特許文献】**

50

【 0 0 0 6 】

【特許文献 1】米国特許第 6,738,388 号

【発明の概要】

【発明が解決しようとする課題】

【 0 0 0 7 】

従来、モジュールを構成する機能ブロックのすべて、または少なくともいくつかを、フィールドバスセグメントに接続された F F フィールド装置内に配置することによって、そして、制御ループを実行するために、それらのブロックが正確にスケジュールされた方法で通信可能に相互作用することを可能にすることによって、特定の制御モジュールを F F プロトコルの機能を使用して分散して動作するように構成することが知られている。残念なことに、多くの場合、特定の制御モジュールを構成する機能ブロックのすべてが、フィールドバスセグメントに接続された実際のフィールド装置内で実行されることが可能とは限らず、その理由は、機能ブロックの多くがフィールドバスセグメントに接続された特定のフィールド装置によって単にサポートされていなかったためである。これらの場合、機能ブロックのいくつか（通常は、制御計算（例えば、比例、積分、および微分（PID）制御ルーチン）を実施する制御機能ブロック）を、入出力装置を介してフィールドバスセグメントに接続されたコントローラ内に配置することが必要であったか、または標準的に実施されていた。ファウンデーション（FOUNDATION（登録商標））フィールドバス（Fieldbus）は、多数のそのような入出力装置を定義およびサポートし、それらのうちで最も一般的なものは、H1 カードまたは H1 装置と呼ばれる。

10

20

【 0 0 0 8 】

しかし、特定の制御モジュールの機能ブロックが、コントローラ内に配置されて実行される場合、この機能ブロックは、フィールドバスセグメントに接続された F F フィールド装置内に配置された機能ブロックの実行および動作に対して非同期に実行され、その理由は、コントローラは、フィールドバスセグメントに直接には接続されず、したがってフィールドバスセグメントの通信スケジュールに直接には参加しないからである。結果として、この場合、制御モジュール自体の全体的な実行レートは、フィールドバスセグメント内で全体が動作する制御モジュールに比較して遅くなり、これは、（制御モジュールの機能ブロックのうちの 1 つを実行している）コントローラと（制御モジュールの機能ブロックのうちのその他のものを実行している）F F フィールド装置との間の信号の受け渡しを確実なものにするために、コントローラとフィールドバスセグメント上のフィールド装置との間の通信が、入出力装置によってフィールドバスセグメントスケジュールの外部で管理されなければならないという事実による。したがって、残念なことに、F F フィールド装置とコントローラとの両方の中に配置された機能ブロックを有する制御モジュールは、フィールドバスセグメントスケジュールによって定義される 1 つのマクロサイクル内で実行することは一般に不可能であり、その代わりに、コントローラと F F フィールド装置との間の入出力装置を介した適切な通信を可能にするために、フィールドバスセグメントスケジュールの一連のマクロサイクルにわたって動作する必要がある。

30

【 0 0 0 9 】

したがって、現在のところ、制御ループを実施するために必要とされる機能ブロックのいくつかまたはすべては、顧客が購入してプロセス制御システム内にインストールした実際の F F フィールド装置の識別情報に基づいて、フィールドバス装置に割り当てられる場合がある。しかし、特定の制御ループ（または制御モジュール）を実施するために必要とされるブロックのうちの 1 つ以上が、フィールドバス装置に割り当てられない場合、それらのブロックは、フィールドバスセグメント上にはない、コントローラ内で実行される。このようにブロックを分割することは、制御ループ内に大幅な遅延を導入し、なぜなら、コントローラ内のブロックの実行は、フィールドバスセグメント上のブロックの実行と同期されないからである。

40

【課題を解決するための手段】

【 0 0 1 0 】

50

ファウンデーション（FOUNDATION（登録商標））フィールドバスプロトコルを実施するものなどの、通信ネットワークの入出力（I/O）装置が、（例えば、そのI/O装置に接続されたフィールドバスセグメント上の）フィールド装置によってはサポートされない機能ブロックの実行をサポートするように構成される。この機能の結果として、FF H1カードであってもよいI/O装置は、従来はプロセスコントローラ内で実行されなければならなかった制御ブロックまたはその他の機能ブロックを実行することが可能になる。I/O装置の、この追加機能は、I/O装置に割り当てられた機能ブロックが、バルブ、またはトランスミッタ装置などの、従来のフィールドバスフィールド装置には割り当てられていないとはいえ、フィールドバスセグメントの通信スケジュール内に同期式に参加することを可能にする。

10

**【0011】**

さらに、装置または方法は、特定の制御モジュールのクリティカルな機能ブロックのすべてが、ファウンデーション（FOUNDATION（登録商標））フィールドバスフィールド装置に、または、特定のフィールドバスセグメントに関連付けられたI/O装置に、割り当てられることが可能かどうかを判定することによって、フィールドバスセグメント上での同期実行のために制御モジュールを自動的に構成する。割り当てられることが可能である場合、装置または方法は、さもなければコントローラ内で実行されるようにスケジュールされるであろう、制御モジュールの機能ブロックを、フィールドバスセグメントのためのI/O装置に、自動的に割り当てる。この技術は、それにより、制御モジュール内の機能ブロックのすべて（あるいは、少なくとも、制御モジュール内のクリティカルな、または必要な機能ブロックのすべて）が、フィールドバスセグメントの1マクロサイクル内で実行され、それにより、フィールドバスセグメント上で同期して実行されることを可能にする。この動作は、さらに、制御モジュールの全体的な実行レートを増加させる。言い換えると、開示された装置および方法は、例えば制御機能ブロックを含む、制御モジュールの機能ブロックが、フィールドバスセグメントの1つのスケジュールされたマクロサイクル内で、実行され、かつフィールドバスセグメント上で相互に通信することを可能にし、それにより、コントローラと、フィールドバスセグメント上のフィールド装置との間で分割された機能ブロックを有する制御モジュールよりも速い、制御モジュールの動作を可能にする。

20

**【0012】**

開示された装置および方法は、さらに、1つ以上の制御モジュールの機能ブロックが、I/O装置を含む、フィールドバスセグメント上のさまざまな装置に割り当てられる方法に基づいて、フィールドバスセグメントのための通信および実行スケジュールを自動的に生成してもよい。このスケジュールは、フィールドバスセグメント上で動作するさまざまな異なる制御モジュールからの機能ブロックを含んでもよく、そして、さまざまな走査または実行レートを有する、同じ、または異なる制御モジュールからの機能ブロックを含んでもよい。

30

**【発明の効果】****【0013】**

フィールドバスセグメント上の装置への機能ブロックの自動割り当てと、フィールドバスセグメント上の通信および実行スケジュールの自動生成とは、オペレータまたは構成エンジニアが、機能ブロックをどこに割り当てるべきかについて、またはフィールドバスセグメントを構成する際に使用すべき最良のスケジュールについて、心配する必要なしに、特定のフィールドバスセグメント上で実行される制御ループ（制御モジュール）を単に指定することを可能にする。この動作は、それにより、プロセス制御システム資産のより容易かつより最適な使用を可能にし、一方でそれと同時に、特定の制御モジュールに関連付けられた機能ブロックのうちなるべく多くを、フィールドバスセグメント上のI/O装置内に配置することによって、可能な場合にはより高速に実行されるように、制御モジュールを構成することを可能にする。

40

**【図面の簡単な説明】**

50

## 【 0 0 1 4 】

【図 1】フィールドバス I / O 装置を介してフィールドバスセグメントに通信可能に結合された分散型プロセスコントローラを含む、分散型プロセス制御ネットワークのブロック図である。

【図 2】フィールドバスセグメント上での、制御モジュールの機能ブロックの同期した動作を可能にするために、制御モジュールの機能ブロックが、特定のフィールドバスセグメント上のフィールド装置および I / O 装置に割り当てられる方法を示すブロック図である。

【図 3 A】制御モジュールの機能ブロックを、フィールドバスセグメント上のフィールドバス I / O 装置に割り当てるか、またはコントローラ装置に割り当てるかを決定する、例示的な機能ブロック割り当てルーチンの動作を示すフローチャートである。

10

【図 3 B】制御モジュールの機能ブロックを、フィールドバスセグメント上のフィールドバス I / O 装置に割り当てるか、またはコントローラ装置に割り当てるかを決定する、例示的な機能ブロック割り当てルーチンの動作を示すフローチャートである。

【図 4】機能ブロックを割り当てる開示された方法に従って、1 つ以上の制御機能ブロックが、フィールドバスセグメント上の I / O 装置には割り当てられず、しかし、その代わりに、コントローラに割り当てられる、従来の制御モジュールを示すブロック図である。

【図 5】機能ブロックを割り当てる開示された方法に従って、1 つ以上の制御機能ブロックが、フィールドバスセグメント上の I / O 装置には割り当てられず、しかし、その代わりに、コントローラに割り当てられる、従来の制御モジュールを示すブロック図である。

20

【図 6】機能ブロックを割り当てる開示された方法に従って、制御機能ブロックが、フィールドバスセグメント上の I / O 装置に割り当てられる、従来の制御モジュールを示すブロック図である。

【図 7】機能ブロックを割り当てる開示された方法に従って、複数の制御機能ブロックが、フィールドバスセグメント上の I / O 装置に割り当てられる、制御モジュールを示すブロック図である。

【図 8】機能ブロックを割り当てる開示された方法に従って、制御機能ブロックおよび入力 / 出力機能ブロックが、フィールドバスセグメント上の I / O 装置には割り当てられず、しかし、その代わりに、コントローラに割り当てられる、制御モジュールを示すブロック図である。

30

【図 9】機能ブロックを割り当てる開示された方法に従って、複数の制御機能ブロックが、フィールドバスセグメント上の I / O 装置に割り当てられる、制御モジュールを示すブロック図である。

【図 10】機能ブロックを割り当てる開示された方法に従って、制御モジュールの複数の制御機能ブロックが、フィールドバスセグメント上の I / O 装置に割り当てられる、制御モジュールを示すブロック図であり、ここで、いくつかの機能ブロックは、フィールドバスセグメント上のその他の機能ブロックよりも速いレートで実行される。

【図 11】図 10 の制御モジュールの複数の機能ブロックの動作を示す、フィールドバスセグメントのためのマクロサイクルの図であり、ここで、いくつかの機能ブロックは、フィールドバスセグメント上で、その他の機能ブロックよりも速いレートで動作する、または実行される。

40

【図 12】フィールドバスセグメント上で動作している複数の異なる制御モジュールに関連付けられた機能ブロックの動作を示す、フィールドバスセグメントのためのマクロサイクルの図であり、ここで、いくつかの機能ブロックは、その他の機能ブロックとは異なるレートで実行される。

【図 13】フィールドバスセグメントに結合されたコントローラを有する、別の例示的なプロセス制御システムを示すブロック図である。

【図 14】図 13 の例示的なプロセス制御システム内でプロセス制御ルーチンを実施するために使用される機能ブロック、および機能ブロックの間の相互接続の、グラフィカルユーザーインターフェース (GUI) 表示を示す。

50

【図15】図13のプロセス制御システムの、異なるフィールド装置に関連付けられた機能ブロックを、通信バスを介して結合する、例示的な機能ブロック結合構成である。

【図16】2つの連続した500ミリ秒ループの実行中における、図13および図14の機能ブロックのうちのいくつかの、実行の間のタイミング関係を示す、例示的な実行シーケンス図である。

【図17】2つの連続した2000ミリ秒ループの実行中における、図13および図14の機能ブロックのうちのいくつかの、実行の間のタイミング関係を示す、別の例示的な実行シーケンス図である。

【図18】異なるブロック走査レートを有し、かつ、図13の例示的なプロセス制御システムの同じ通信バスに通信可能に結合されたフィールド装置に割り当てられた、機能ブロックを実行するための方法に従って実施される、例示的な実行シーケンス図である。

【図19】機能ブロックが、それらのブロック走査レートと、それぞれのループ実行周期とを使用して実行されることを可能にする、本明細書に記載された例示的な方法および装置に従って実施される、例示的な実行シーケンス図である

【図20】機能ブロックのブロック未加工実行周期を切り上げるために使用されてもよい、例示的な丸めテーブルである。

【図21】図13および図14の機能ブロックによって、それぞれのサブスケジュールの間の実行のために必要とされる、開始時間オフセットの数量を示す、例示的な、開始時間オフセットの数量テーブル900である。

【図22】複数のフィールド装置がプロセス制御システムの通信バスに通信可能に結合され、それぞれの機能ブロックを実行するように構成された、別の例示的な機能ブロック構成を示す。

【図23】図22の例示的な機能ブロック構成の、それぞれの機能ブロックについて必要とされる、開始時間の数量を示す、別の例示的な、開始時間オフセットの数量テーブルを示す。

【図24】本明細書に記載された例示的な方法および装置を使用して生成される、図22の機能ブロックに関連付けられたスケジュールを示す、別の例示的な実行シーケンス図である。

【図25】図13のフィールド装置と、図13の機能ブロックと、機能ブロックに対応する開始時間オフセットとを表すオブジェクトの間の関係を示す、例示的なオブジェクト指向ソフトウェア実行環境である。

【図26】図22のフィールド装置と、図22の機能ブロックと、機能ブロックに対応する開始時間オフセットとを表すオブジェクトの間の関係を示す、別の例示的なオブジェクト指向ソフトウェア実行環境である。

【図27】図22の機能ブロックに関連付けられた、図26の開始リストが、ワークステーションから、図22の対応するフィールド装置にコピーされる、例示的な方法を示す。

【図28】本明細書に記載された例示的な方法および装置を使用して、図19および図24のスケジュールを生成するように構成された、複数のクラスを有する、例示的なオブジェクト指向プログラミング環境を示す。

【図29】本明細書に記載された例示的な方法に従ってスケジュールを生成するために使用されてもよい、例示的な装置の詳細なブロック図である。

【図30】本明細書に記載されたようにスケジュールを生成するための、図29の例示的な装置を実施するために使用されてもよい、例示的な方法のフロー図である。

【図31】機能ブロックによって必要とされる、開始時間オフセットの数量を決定するために、図30の例示的な方法に関連して実施されてもよい、例示的な方法のフロー図である。

【図32A】サブスケジュールを生成するため、および生成されたサブスケジュールに基づいてスケジュールを生成するために、図30の例示的な方法に関連して実施されてもよい、例示的な方法のフロー図である。

【図32B】サブスケジュールを生成するため、および生成されたサブスケジュールに基

10

20

30

40

50

づいてスケジュールを生成するために、図 30 の例示的な方法に関連して実施されてもよい、例示的な方法のフロー図である。

【図 33】機能ブロックについての開始時間オフセット値を決定するために、図 32 の例示的な方法に関連して実施されてもよい、例示的な方法のフロー図である。

【図 34】図 24 のスケジュールの、例示的な機能実行およびデータ転送シーケンス図である。

【図 35】本明細書に記載された例示的なシステムおよび方法を実施するために使用されてもよい、例示的なプロセスシステムのブロック図である。

【発明を実施するための形態】

【0015】

ここで、図 1 を参照すると、プロセス制御システム 10 は、データヒストリアン 12 と、それぞれが表示画面 14 を有する 1 つ以上のホストワークステーションまたはコンピュータ 13 (任意のタイプのパーソナルコンピュータ、ワークステーションなどであってもよい) とに通信可能に接続された、プロセスコントローラ 11 を含む。コントローラ 11 は、さらに、入出力 (I/O) 装置またはカード 26 および 28 を介して、フィールド装置 15 ~ 22 に接続される。データヒストリアン 12 は、データを記憶するための、任意の所望のタイプのメモリと、任意の所望のまたは公知のソフトウェア、ハードウェア、またはファームウェアとを有する、任意の所望のタイプのデータ収集ユニットであってもよく、及び、(図 1 に示すように) ワークステーション 13 とは別個であるか、またはワークステーション 13 のうちのいずれかの一部であってもよい。例として、エマーソンプロセスマネジメント (Emerson Process Management) によって販売されているデルタ V (Delta V (登録商標)) コントローラであってもよい、コントローラ 11 は、例えば、イーサネット (登録商標) 通信リンクまたは任意のその他の所望の通信ネットワーク 29 を介して、ホストコンピュータ 13 およびデータヒストリアン 12 に通信可能に接続される。通信ネットワーク 29 は、ローカルエリアネットワーク (LAN)、ワイドエリアネットワーク (WAN)、電気通信ネットワーク、インターネット (例えば、ワールドワイドウェブ) などの形態であってもよく、そして、ハードワイヤードおよび/またはワイヤレス技術を使用して実施されてもよい。コントローラ 11 は、例えば、標準的な 4 ~ 20 mA 装置、および/または任意のスマート通信プロトコル (ファウンデーション (FOUNDATION (登録商標)) フィールドバス (Fieldbus) プロトコル、ハート (HART) プロトコル、プロフィバス (Profibus) プロトコル、あるいはプロセス制御業界で使用される任意のその他の周知のまたは標準的な通信プロトコルなど) と関連付けられた、任意の所望のハードウェアおよびソフトウェアを使用して、フィールド装置 15 ~ 22 に通信可能に接続される。

【0016】

フィールド装置 15 ~ 22 は、センサ、バルブ、トランスミッタ、ポジショナなどの、任意のタイプの装置であってもよく、一方、I/O カード 26 および 28 は、任意の所望の通信プロトコルまたはコントローラプロトコルに準拠するか、あるいはそれらのプロトコルをサポートする、任意のタイプの I/O 装置であってもよい。図 1 に示す実施形態では、フィールド装置 15 ~ 18 は、アナログ回線 (またはアナログ回線とデジタル回線の組み合わせ) を介して I/O カード 26 との通信を行う、標準的な 4 ~ 20 mA 装置またはハート (HART) 装置であり、一方、フィールド装置 19 ~ 22 は、ファウンデーション (FOUNDATION (登録商標)) フィールドバス (Fieldbus) プロトコル通信を使用して、デジタルバス 28A を介して I/O カード 28 との通信を行う、ファウンデーション (FOUNDATION (登録商標)) フィールドバス (Fieldbus) (FF) 装置として示されている。この場合、I/O カード 28 は、FF プロトコルをサポートするフィールドバスポートを有する H1 カードである。理解されるように、デジタルバス 28A と、それに接続されたフィールド装置 19 ~ 22 と、I/O カード 28 とは、フィールドバスセグメントを形成する。もちろん、所望される場合には、I/O カード 28 は、他のフィールドバスセグメント (図示せず) に接続されてもよく、フィ

10

20

30

40

50

ールドバスセグメントは、それに接続された、FFプロトコルをサポートする任意の数およびタイプの装置を含んでもよい。あるいは、フィールド装置15~22は、任意のその他の公知のフィールドバスプロトコル、あるいは、現在知られている、または将来開発される可能性がある、任意のその他の通信規格またはプロトコルを含む、任意のその他の所望の規格またはプロトコルに準拠してもよい。

#### 【0017】

コントローラ11は、プラント10内の多数の分散型コントローラのうちの1つであってもよく、それらの分散型コントローラのそれぞれは、通信リンク29に接続されてもよく、かつ、それらの分散型コントローラのそれぞれは、FFプロトコルをサポートするI/O装置(本明細書では、FF I/O装置またはフィールドバスI/O装置とも呼ばれる)を介して、1つ以上のフィールドバスセグメントに接続されてもよい。コントローラ11は、少なくとも1つのプロセッサをその中に含み、工業プロセスプラント内の制御ループを実行する1つ以上のプロセス制御モジュールを、実施または監視するように動作する。制御モジュールは、複数の通信可能に相互接続された機能ブロックを含んでもよく、それらの機能ブロックは、コントローラ11の中に、および/またはフィールド装置19~22のうちの一つかの中に、および、以下で述べるようにI/O装置28の中に記憶されて、それらの中で実行されてもよい。コントローラ11は、任意の所望の方法でプロセスを制御および監視するために、装置15~22、ホストコンピュータ13、およびデータヒストリアン12と通信する。本明細書に記載された任意の制御ルーチンまたは要素は、所望される場合には、さまざまなコントローラまたはその他の装置によって実施または実行される、その部分を有してもよいということに留意すべきである。同様に、プロセス制御システム10内で実施される、本明細書に記載された制御ルーチンまたは要素は、ソフトウェア、ファームウェア、ハードウェアなどを含む、任意の形態を取ってもよい。モジュールであってもよく、または、サブルーチン、(コード行などの)サブルーチンの部分などの、制御手順の任意の部分であってもよい、制御ルーチンは、ラダーロジック、シーケンシャルファンクションチャート、機能ブロック図、オブジェクト指向プログラミング、あるいは、任意のその他のソフトウェアプログラミング言語または設計パラダイムを使用するなどの、任意の所望のソフトウェア形式で実施されてもよい。同様に、制御ルーチンは、例えば、1つ以上のEPROM、EEPROM、特定用途向け集積回路(ASIC)、または任意のその他のハードウェアまたはファームウェア要素の中にハードコートされてもよい。さらに、制御ルーチンは、グラフィカル設計ツールまたは任意のその他のタイプのソフトウェア/ハードウェア/ファームウェアプログラミングまたは設計ツールを含む、任意の設計ツールを使用して設計されてもよい。したがって、コントローラ11は、本明細書に具体的に記載された手法以外の任意の所望の手法で、制御戦略または制御ルーチンを実施するように構成されてもよい。

#### 【0018】

一実施形態では、コントローラ11は、プロセス制御モジュールと一般に呼ばれるものを使用する制御戦略(ストラテジー)を実施し、プロセス制御モジュールのそれぞれは、相互接続された機能ブロックの組から構成されてもよい。この場合、各機能ブロックは、全体的な制御モジュールの一部、または全体的な制御モジュール内のオブジェクトであり、プロセス制御システム10内のプロセス制御ループを実施するために(リンクと呼ばれる通信を介して)他の機能ブロックと協働して動作する。機能ブロックは、一般に、トランスミッタ、センサ、またはその他のプロセスパラメータ測定装置に関連付けられるものなどの、入力機能、あるいは、比例/積分(PI)、比例/微分(PD)、比例/積分/微分(PID)、ファジーロジックなどの制御を実行する制御ルーチンに関連付けられるものなどの、制御機能、あるいは、プロセス制御システム10内の何らかの物理的機能を実行するためにバルブなどの何らかの装置の動作を制御する、出力機能のうちの一つを実行する。もちろん、ハイブリッドおよびその他のタイプの機能ブロックが存在する。機能ブロックは、コントローラ11内に記憶されて、コントローラ11によって実行されてもよく、これは、それらの機能ブロックが、標準的な4~20mA装置、ならびに、ハート

10

20

30

40

50

装置および場合によってはフィールドバス装置などの、何らかのタイプのスマートフィールド装置のために使用される場合、またはそれらの装置に関連付けられる場合に一般的に当てはまる。しかし、場合によっては、それらの機能ブロックは、フィールド装置自体の中に記憶されて、フィールド装置自体によって実行されてもよく、これは、フィールドバス装置の場合に当てはまる可能性がある。制御システムの説明は、本明細書では、オブジェクト指向プログラミングパラダイム使用する機能ブロック制御戦略を用いて提供されているが、制御戦略、あるいは制御ループまたはモジュールは、ラダーロジック、シーケンシャルファンクションチャートなどの、その他の規定を使用して、あるいは、任意のその他の所望のプログラミング言語またはパラダイムを使用して、実施または設計されてもよい。

10

**【 0 0 1 9 】**

図1の拡大ブロック30によって示すように、コントローラ11は、プロセス制御モジュール32および34として図1に示されている複数の単一入力/単一出力制御ループを監視するか、またはそれらの実施に関与してもよく、そして、プロセス制御モジュール36によって示されている1つ以上のアドバンスト制御ループを実施してもよい。各制御ループを構成する機能ブロックは、一般に、制御モジュールと呼ばれ、それらの機能ブロックは、従来、コントローラ11内、かつ/または、フィールドバスセグメント28Aを使用する場合は、フィールド装置15~22内に記憶されていた。制御モジュール32および34は、適切なアナログ入力(AI)およびアナログ出力(AO)機能ブロックに接続された、それぞれ、単一入力/単一出力ファジーロジック制御ブロックおよび単一入力/単一出力PID制御ブロックを使用して、単一ループ制御を実行するものとして示されている。AIおよびAO機能ブロックは、バルブなどのプロセス制御装置と関連付けられてもよく、温度トランスミッタおよび圧力トランスミッタなどの測定装置と関連付けられてもよく、または、プロセス制御システム10内の任意のその他の装置と関連付けられてもよい。アドバンスト制御ループ36は、多数のAI機能ブロックに通信可能に接続された入力と、多数のAO機能ブロックに通信可能に接続された出力とを有する、アドバンスト制御ブロック38を含むものとして示されているが、アドバンスト制御ブロック38の入力および出力は、その他のタイプの入力を受信するため、およびその他のタイプの制御出力を提供するために、任意のその他の所望の機能ブロックまたは制御要素に通信可能に接続されてもよい。所望される場合には、アドバンスト制御ブロック38は、プロセスまたはプロセスの部分の最適制御を実行するための、モデル予測制御ルーチンを最適化と統合した制御ブロックであってもよい。アドバンスト制御ブロック38を含む、図1に示す制御モジュールおよび機能ブロックは、従来、コントローラ11によって実行されていたか、あるいは、ワークステーション13のうちの1つ、またはさらには、フィールド装置19~22のうちの1つなどの、その他の処理装置内に配置されて、それらの処理装置内で実行されていたということが理解されるであろう。しかし、以下でさらに詳細に述べるように、それらの機能ブロックは、さらに、有利には、フィールドバスセグメントの部分として同期して動作するように、I/O装置28などのI/O装置内に記憶されてもよい。

20

30

**【 0 0 2 0 】**

図1に示すように、ワークステーション13のうちの1つは、コントローラ11内の制御モジュール、特に、バス28Aによって定義されるフィールドバスセグメントを使用するものを作成、割り当て、ダウンロード、および実施するために使用される制御ブロック割り当ておよびスケジューリングルーチンを含む。制御ブロック割り当ておよびスケジューリングルーチン40は、ワークステーション13内のメモリ内に記憶されて、ワークステーション13内のプロセッサによって実行されてもよいが、所望される場合には、このルーチン(またはその任意の部分)は、それに加えて、または別法として、プロセス制御システム10内の任意のその他の装置内に記憶されて、その装置によって実行されてもよい。一般的に言えば、ルーチン40は、作成された制御モジュール内の機能ブロックを、実行のために、プロセスプラント内の特定の装置に割り当てるように動作するルーチン4

40

50

2を含み、ルーチン42は、より詳細には、(バス28Aによって定義されるセグメントなどの)特定のフィールドバスセグメントのために作成された各制御モジュールの機能ブロックが、セグメント上の装置に、またはセグメントを動作させるコントローラに割り当てられる方法を、自動的に最適化するように動作する。

#### 【0021】

さらに、所望される場合、ルーチン40は、ルーチン42によって、セグメント上のフィールドバスフィールド装置に割り当てられた機能ブロックと、セグメントのためのI/O装置に割り当てられた機能ブロックとに基づいて、フィールドバスセグメントの通信スケジュール(すなわち、マクロサイクル)を自動的に作成または構成する、ルーチン44を含んでもよい。ルーチン44は、ルーチン42によって実行された割り当てに基づいて、特定のフィールドバスセグメントのための通信および実行スケジュールをユーザが最適に生成するのを容易にする。1つの場合には、ルーチン44は、フィールドバスセグメントのためのマクロサイクルまたはスケジュールを、従来技術において一般的に行われているようにこのセグメントに割り当てられたモジュールの最も遅い実行レートに基づいて生成するのではなく、セグメントに割り当てられたモジュールの最も速いモジュール実行レートに基づいて生成してもよい。この場合、ルーチン44は、インタフェース装置13を介してユーザによって指定された機能ブロックマルチプライヤが、どのブロック実行がスケジュール内でスキップされるべきかを指示することを可能にしてもよく、これは、フィールドバスセグメント上の帯域幅を節約するのに役立つ。さらに、所望される場合、ルーチン40は、情報を入力および表示するため、ならびにルーチン42および44の性能を指示するために、ユーザがルーチン42および44と対話することを可能にする、ルーチン46を含んでもよい。

#### 【0022】

上述のように、FF装置をサポートする公知のコントローラシステムは、ユーザがFF装置を使用することをより容易にするが、特定の制御モジュールの制御経路内の機能ブロックが、コントローラ(例えば、図1のコントローラ11)と、フィールドバスセグメント上のフィールドバス装置(例えば、図1の装置19~22)との間で分割される場合、制御ループ内に大幅な遅延が導入される可能性がある。さらに、セグメント上のすべてのブロックが同じレートで実行されるという従来技術の要件は、同じフィールドバスセグメント上のFF装置内で複数のループが実施される際に、場合によっては制御性能に影響する可能性があり、なぜなら、場合によっては、異なる制御ループまたは特定の制御ループ内のブロックは、異なるレートで動作するように構成される可能性があるからである。ルーチン40は、一般に、コントローラ内制御(control in the controller)(CIC)と比較して、フィールド内制御(Control in the field)(CIF)を使用するために必要とされる技術的労力を減少させる方法で、そして、CICをフィールド装置と組み合わせて使用する場合に制御ループ内に導入される変動および総合遅延を最小にする方法で、フィールドバスセグメントに接続された装置に制御ブロックを割り当てるように動作する。さらに、ルーチン40は、CIFのための追加の機能ブロックの選択肢をユーザに提供するように動作してもよく、そして、フィールドバスセグメント上で実施される1つ以上のモジュールの、全体的により速い動作を可能にする、フィールドバスセグメントのためのスケジュールを定義してもよい。

#### 【0023】

より詳細には、ルーチン42の機能および動作をサポートするために、フィールドバスセグメントに接続されたI/O装置(すなわち、図1のI/O装置28)は、従来はコントローラ11内で実施または実行されてきた機能ブロックのサブセットを実行するように構成される。特に、フィールドバスポートが、I/O装置のために指定または作成され、このポートは、フィールドバスセグメントに接続された独立したフィールドバス装置としてモデル化されてもよい。各フィールドバスポートは、したがって、ポート(または、一般にI/O装置)によってサポートされる機能ブロック、開始リストエントリの数その他を指定する、フィールドバス装置リビジョンファイルを含んでもよい。各フィールドバス

10

20

30

40

50

ポートは、プロセス制御システムとの関連で、その他の装置に類似した方法で自動的に生成されてもよい、グローバルシステム装置タグを有してもよい。同様に、フィールドバスポートに関する参照メニューオプションが、そのポートに割り当てられる、参照されるモジュール/ブロックのすべてをリストするために提供されてもよい。しかし、一般的に言えば、フィールドバスポートは、FFフィールド装置では有するのが一般的である、リソースブロックまたはトランスデューサブブロックは有さない。一実施形態では、各フィールドバスポートは、最大32の機能ブロックをサポートすることが可能である。しかし、サポートされるブロックの実際の数およびタイプは、装置内で得られる開始リストエントリの数によって決定される。さらに、各フィールドバスポートは、以下でさらに詳細に説明する、ユーザ設定可能な「自動計算スケジュールマクロサイクル (auto-calculate schedule macro cycle)」プロパティを有してもよい。

10

**【0024】**

フィールドバスセグメントに接続されたI/O装置がこの追加機能を含む場合、ルーチン42は、機能ブロックをI/O装置(例えば、図1のI/O装置28)のフィールドバスポート内で実行されるように割り当てることが可能である。一般的に言えば、ルーチン42が、制御モジュールの機能ブロックをフィールドバスセグメントのためのI/O装置のフィールドバスポート内で実行されるように割り当てるのは、それらの機能ブロックが、I/O装置内でサポートされる機能ブロックのサブセットのメンバーであり、かつ、フィールドバスセグメント上で実行され、またはフィールドバスセグメントを使用する、特定の制御モジュールの、すべてのフィールドバス入力機能ブロックとフィールドバス出力機能ブロックとの間の主要制御経路内にある場合である。このようにして、ルーチン42は、可能な場合、制御モジュールの主要制御経路内のすべての機能ブロックが、フィールドバスセグメントに接続されている装置のうちの1つにおいて実行されることを確実にし、それにより、それぞれのそのような機能ブロックがフィールドバスセグメント上のマクロサイクルに直接参加する(動作および通信のためにスケジュールされる)ことを可能にする。この割り当て技術は、フィールドバスセグメント内で制御モジュール(すなわち、制御ループ)が実施されるレートを、コントローラ11と、フィールドバスセグメント上の1つ以上のフィールドバス装置との間で分割された機能ブロックを有するプロセスモジュールが動作するレートに比較して、速める、即ち増加させるように機能する。

20

**【0025】**

ここで、図2を参照すると、制御モジュール50(モジュール1と称される)は、AO(アナログ出力)機能ブロック51と、AI(アナログ入力)機能ブロック52と、PID機能ブロック53とを含むものとして示されている。制御モジュール50は、例えば、図1のモジュール34であってもよい。図2の点線で示されているように、AO機能ブロック51は、フィールドバスフィールド装置54内で動作するように(通常は、オペレータまたは構成エンジニアによって)割り当てられ、一方、AI機能ブロック52は、別のフィールドバスフィールド装置55内で動作するように(同様に、通常は、オペレータまたは構成エンジニアによって)割り当てられる。この場合、AOブロック51は、バルブ54内で実行されるように割り当てられ、一方、AIブロックは、センサまたはその他の機器55内で実行されるように割り当てられる。重要なことに、フィールド装置54および55は、共通フィールドバスセグメント56に接続されているか、またはその部分である。

30

40

**【0026】**

一般に、従来技術のシステムでは、PID機能ブロック53は、フィールドバス装置54または55のいずれによってもサポートされず、したがって、PID機能ブロック53は、(図1のコントローラ11であってもよい)コントローラ57内で実行されるように、オペレータまたは構成エンジニアによって割り当てられる。この従来の実施は、図2の点線57Aによって示されている。しかし、より長い点線59Aによって図2に示されているように、ルーチン42はここで、PID機能ブロック53を、I/O装置28のフィールドバスポート59内での実行のために、FFプロトコルをサポートするI/O装置5

50

8 ( 図 2 で、フィールドバスカードとしてラベル付けされている ) 内での実行のために割り当ててもよい。図 2 に示すように、フィールドバスポート 5 9 は、フィールドバスセグメント 5 6 に接続され、したがって、フィールドバスセグメント 5 6 の通信スケジュールに直接参加することが可能である。PID 機能ブロック 5 3 を、コントローラ 5 7 ではなく、フィールドバスポート 5 9 に割り当てることにより、ブロック 5 1、5 2、および 5 3 のそれぞれは、フィールドバスセグメント 5 6 上の装置内に配置され、したがって、フィールドバスセグメント 5 6 によって使用されるマクロサイクル内での動作のために、直接スケジュールされることが可能となる。この割り当ての結果として、機能ブロックのうちのいくつか ( すなわち、機能ブロック 5 1 および 5 2 ) をフィールドバスセグメント 5 6 上で実行させ、かつ、機能ブロックのうちのいくつか ( すなわち、機能ブロック 5 3 ) をフィールドバスセグメント 5 6 の外部 ( すなわち、コントローラ 5 7 内 ) で実行させるということなしに、モジュール 5 0 全体がフィールドバスセグメント 5 6 上で動作するようになる。

#### 【 0 0 2 7 】

所望される場合、機能ブロックが I / O 装置のフィールドバスポートに割り当てられる際に、機能ブロックはコントローラ 5 7 内で作成されてもよく、そして、シャドウ機能ブロックという周知の概念を使用して、フィールドバスポート 5 9 内の機能ブロックにリンクされてもよい。特に、コントローラは、コントローラ 5 7 内のシャドウ機能ブロックを使用することによって、I / O 装置のフィールドバスポートに割り当てられた機能ブロックと通信し、それらの装置に情報を送信すること、およびそれらの装置から情報を取得することが可能である。公知のように、シャドウ機能ブロックは、アルゴリズムを実行しないブロックであり、そして従来は、フィールドバス装置内で実行されるフィールドバスブロックと通信するために、コントローラベースのモジュールによって使用されている。シャドウ機能ブロックの構成および使用の、より完全な説明は、米国特許第 6, 7 3 8, 3 8 8 号明細書に記載されており、当該特許は参照により本明細書に明示的に援用される。それゆえ、この特徴については、本明細書ではさらに詳細には説明しない。

#### 【 0 0 2 8 】

一般的に言えば、図 1 のルーチン 4 2 は、図 2 のフィールドバスポート 5 9 またはコントローラ 5 7 のいずれかに機能ブロックを割り当てて、それにより、可能な最適な方法でこの割り当てを行うことにおける、一定の規則の組を実施してもよい。一実施形態では、ルーチン 4 2 は、特定の制御モジュールのクリティカルフロー経路内のすべての機能ブロックが、特定のフィールドバスセグメントに接続された装置内で実行されることが可能かどうかを判定し、可能である場合、フィールドバスセグメント上のフィールド装置には割り当てることができない機能ブロックを、そのセグメント上の I / O 装置のフィールドバスポートに割り当てる。可能ではない場合、ルーチン 4 2 は、フィールドバスセグメント上のフィールド装置によってサポートされない機能ブロックを、セグメントに関連付けられたコントローラに割り当てる。このようにして、ルーチン 4 2 は、I / O 装置内への機能ブロックの割り当てを、制御モジュール全体がフィールドバスセグメント上で実行可能であることを保証するために必要な割り当てに制限し、それにより、I / O 装置の、特に I / O 装置のフィールドバスポートのリソースを節約する。

#### 【 0 0 2 9 】

特定のモジュールがフィールドバスセグメント上の I / O 装置に自動的に割り当てられるべきかどうかを、ユーザまたは構成エンジニアが指定することを可能にするために、「自動計算スケジュールマクロサイクル」と称されるユーザ設定可能なプロパティが、フィールドバスポートに追加されてもよい。ルーチン 4 2 がこのプロパティが設定されていると判定した場合、ルーチン 4 2 は、I / O 装置のフィールドバスポートを使用して、フィールドバスセグメント上での実行のために、モジュールをスケジュールすることを試みる。さらに、ユーザは、ルーチン 4 2 が、特定のモジュールの機能ブロックを、フィールドバスセグメントのための I / O 装置のフィールドバスポートに割り当てようと試みるべきかどうかをモジュールごとに指定してもよい。特に、各制御モジュールは、例えば、「フ

10

20

30

40

50

フィールドバスポートへの機能ブロックの自動割り当て (auto-assign function blocks to Fieldbus port)」と称される、ユーザ設定可能なプロパティを有してもよく、このプロパティは、設定されている場合、モジュールの機能ブロックが、フィールドバスセグメントのための I/O 装置のフィールドバスポートに割り当てられることを可能にする。本明細書では「自動割り当て (auto-assign)」プロパティとも呼ばれる、このプロパティは、機能ブロックレベルではなく、モジュールレベルで有効にされ、その理由は、ルーチン 42 は、機能ブロックを実行のためにフィールドバス I/O 装置に割り当てようと試みるどうかを、機能ブロックごとではなくモジュールごとに判定するからである。したがって、一般的に言えば、自動割り当てプロパティはモジュールごとにオンおよびオフにされるべきであり、このプロパティは制御モジュールの最高レベルにおいてのみ適用されるべきである。

10

**【0030】**

一実施形態では、フィールドバスプロトコルをサポートする I/O 装置 (フィールドバス I/O 装置とも呼ばれる) のフィールドバスポートに機能ブロックを割り当てる場合、ルーチン 42 は、制御モジュールのクリティカルフロー経路内のすべての入力および出力機能ブロックが、共通フィールドバスセグメント上のフィールド装置に割り当てられるようにスケジュールされるかどうかを判定する。そうである場合、ルーチン 42 は、それらのブロックが、フィールドバスセグメント上のフィールド装置には割り当てられないことを前提として、かつ、この割り当てを可能にするための十分なリソースがまだ存在していることを前提として、(入力機能ブロックと出力機能ブロックとの間に配置された) 制御モジュールのクリティカルフロー経路内のその他の機能ブロックを、フィールドバスセグメント上の I/O 装置のフィールドバスポートに割り当てる。例えば、フィールドバス I/O 装置のフィールドバスポートが、制御モジュール内で使用される特定のタイプの制御機能ブロックをサポートしない場合、または、フィールドバスポートに、モジュールの機能ブロックのうちの 1 つを割り当てるための十分なリソースさえも残されていない場合、そのモジュールの機能ブロックのうちのいずれも I/O 装置のフィールドバスポートに割り当てられない。他方、制御モジュールのクリティカルフロー経路内の入力または出力機能ブロックのうちの 1 つ以上が、セグメント上のフィールド装置内で実行されるようにスケジュールされない場合、ルーチン 42 は、そのモジュールの制御機能ブロックおよびその他の機能ブロックを、I/O 装置のフィールドバスポートにではなくコントローラに割り当てる。

20

30

**【0031】**

したがって、一般的に言えば、制御モジュールの機能ブロックが、フィールドバス I/O 装置のフィールドバスポートに割り当てられるのは、(1) モジュールのクリティカルフロー経路内のすべての入力および出力ブロックが、共通フィールドバスセグメント上の装置およびアクチュエータに割り当てられ、かつ、(2) 入力ブロックと出力ブロックとの間のすべての機能ブロックが、フィールドバスポートに割り当て可能である (フィールドバスポートによってサポートされる) 場合である。言い換えると、それらのブロックが、次の条件で、セグメントのためのフィールドバスポートに自動的に割り当てられることにより、モジュールのクリティカルフロー経路内にある機能ブロックは、セグメント上で実行されるように自動的に割り当てられる。(1) モジュールについての自動割り当てプロパティが有効にされている、(2) セグメント上の装置および/またはアクチュエータに割り当てられる、モジュールの、少なくとも 1 つの入力ブロックと少なくとも 1 つの出力ブロックとが存在する、(3) 同期の破れがない (例えば、入力機能ブロックと出力機能ブロックとの間の、クリティカルフロー経路内の各機能ブロックが、フィールドバスポートに割り当てられることが可能である)、かつ (4) フィールドバス I/O 装置のフィールドバスポートについて、「自動計算スケジュールマクロサイクル」プロパティがセットされている。

40

**【0032】**

ここで、一実施形態では、制御ループ (制御モジュール) を通したフォワード経路 (1

50

つまたは複数)のみが、クリティカルフロー経路の部分であるとみなされ、フィードバック経路はいずれも、クリティカルフロー経路の部分であるとはみなされない、ということに留意すべきである。さらに、フィールドバスポートへのブロックの自動割り当ては、制御経路内のI/Oブロックが、同じフィールドバスセグメントに割り当てられること、および、クリティカルフロー経路内のすべての機能ブロックが、フィールドバスポート内でサポートされることを前提とする。一実施形態では、制御モジュールのためにフィールドバスセグメント上のフィールドバス装置に割り当てられる、少なくとも1つの入力ブロックと少なくとも1つの出力ブロックとが存在する限り、クリティカルフロー経路内にAI、AO、DI、またはDOブロックが複数存在する場合、それらのうちの1つのみがフィールドバス装置に結合される必要がある。しかし、この場合、フィールドバス装置に割り  
10  
当てられない入力および出力機能ブロックは、フィールドバスポートには割り当てられず、したがって、それらの機能ブロックは非同期に動作する。ただし、所望される場合、いくつかの実施形態では、入力/出力機能ブロック(例えば、AI、AO、DI、DO、MAI機能ブロック)は、フィールドバスポートに割り当てられてもよく、または、フィールドバスポートには割り当てられないようにされてもよい。他の実施形態では、所望される場合、クリティカルフロー経路内のその他の機能ブロックを、そのセグメントのためのフィールドバスI/O装置のフィールドバスポートに割り当てるために、ルーチン42は、モジュールのクリティカルフロー経路上のすべての入力および出力機能ブロックが、セグメント上のフィールドバス装置に割り当てられるべきであるとしてもよい。

#### 【0033】

したがって、一般的に言えば、モジュールの機能ブロックをフィールドバスポートに割り当てる前に、ルーチン42は、(1)モジュールについての、自動割り当てプロパティが有効にされているかどうか、(2)フィールドバスポート上で、自動計算スケジュールマクロサイクルプロパティが有効にされているかどうか、(3)I/Oカードは、モジュールに関連付けられた機能ブロックをサポートするための十分なリソースを有するかどうか、(4)モジュールのフォワードフロー経路内の、入力機能ブロックと出力機能ブロックとの間のすべての機能ブロックが、フィールドバスカードによってサポートされるかどうか、および(5)フォワードフロー経路内のすべての機能ブロックが同じモジュール内にあるかどうか、を判定してもよい。これらの条件のうちのいずれかが満足されない場合、  
20  
ルーチン42は、フィールドバスセグメント上のフィールドバスI/O装置のフィールド  
30  
バスポートには、モジュールの機能ブロックのうちのいずれも割り当てないことを決定してもよく、しかし、その代わりに、それらの機能ブロックを、フィールドバスセグメントに関連付けられたコントローラに割り当ててもよい。

#### 【0034】

この技術では、ユーザは一般に、フィールドバスI/Oカードのフィールドバスポートから個々の機能ブロックを割り当て解除することはできず、その理由は、機能ブロックはモジュールのプロパティに基づいてこのカードに割り当てられるからである。その代わりに、ユーザは、モジュールの自動割り当てプロパティを設定することによって、モジュールを全体として、自動割り当てすることまたは自動割り当て解除することのみが可能である。  
40

#### 【0035】

所望される場合、制御モジュールの自動割り当てプロパティをユーザがオン/オフした場合、変化を見て結果を明らかにするためにはユーザがモジュールを保存しなければならないことをユーザに通知するために、メッセージがユーザに提供されてもよく、その理由は、ルーチン42の割り当て動作は、モジュールが保存される際に自動的に発生してもよいからである。さらに、各機能ブロックは、このブロックが自動割り当てのために設定されているかどうかを示す、視覚的プロパティを含むことが望ましい。所望される場合、モジュールの自動割り当てプロパティが自動割り当てであるように設定された場合、フィールドバスポートを有するフィールドバスI/O装置にモジュールの機能ブロックが実際に割り当てられるのを妨げるリソースの制限があるならば、ユーザに警告が発行されてもよ  
50

い。例えば、フィールドバスI/Oカードがその限界に達しており、かつ、モジュールが保存される場合、ユーザは、モジュールの機能ブロックは自動割り当てされず、モジュールは自動割り当てプロパティがオフにされて保存されるという警告を受信してもよい。この場合、ブロック自動割り当て機能を動作させるには、ユーザは、他のモジュールの自動割り当てプロパティを解消または設定解除してフィールドバスポート上のいくつかの割り当て空間を空け、モジュールの自動割り当てプロパティをオンにし、次に、モジュールを再保存しなければならない可能性がある。

【0036】

所望される場合、ルーチン42は、フィールドバス装置に割り当てられることが可能な、完全なクリティカルフロー経路を有する潜在的なモジュールのすべてと、それらのモジュールが有する自動割り当てプロパティが「オン」にされているか「オフ」されているかとを、例えばデータベース12内に配置するユーティリティを含んでもよい。このユーティリティは、特定のモジュールについての自動割り当てプロパティをオンまたはオフにすることがリソース空間の作成に役立つ可能性がある場合を、ユーザが理解するのを助けるために使用されてもよい。このユーティリティは、さらに、構成のダウンロードに先立って自動割り当てプロパティを設定することが有利であるモジュールの識別情報を、ユーザが判定するために役立つ場合もある。

【0037】

図3は、ルーチン42が、制御モジュールの機能ブロックをフィールドバスI/O装置のフィールドバスポートに、または特定のフィールドバスセグメントに関連付けられたコントローラに自動的に割り当てるために動作する、さらなる例示的な方法を説明するフローチャート60を示す。ブロック61において、ルーチン42は、調べるべきさらなる制御モジュールがあるかどうかを判定し、ない場合は終了する。調べられるべき1つ以上の制御モジュールがまだ残されている場合、ブロック62で、調べるべき次の制御モジュールを見つける。ブロック63で、次に、選択された制御モジュールについての自動割り当てプロパティが「オン」に設定されているかどうかを判定し、「オン」に設定されていない場合、制御はブロック64に与えられ、ブロック64は、従来技術のシステムにおいて通常行われるように、制御モジュールの未割り当ての制御機能ブロックおよび入力/出力機能ブロックを、コントローラに割り当てるように動作する。しかし、ブロック63で、選択された制御モジュールの自動割り当てプロパティが「オン」に設定されていると判定した場合は、ブロック65で、制御モジュールのクリティカルフロー経路を決定する。上述の説明から理解されるように、クリティカルフロー経路は、入力ブロックから出力ブロックまでの、制御モジュールを通したフォワード経路(1つまたは複数)の各構成要素を含む。このクリティカルフロー制御経路はモジュールを通して進み、それぞれの入力ブロックからそれぞれの出力ブロックまでのフォワードフローリンクによって形成される経路上に配置された、それぞれの機能ブロックを識別することによって決定されてもよい。

【0038】

次に、ブロック66で、制御モジュールのクリティカルフロー経路上の入力機能ブロックおよび出力機能ブロックのそれぞれが、共通フィールドバスセグメント上のフィールドバスフィールド装置に割り当てられるかどうかを判定する。代替の実施形態では、ブロック66で、モジュールのそれぞれのクリティカルフロー経路構成要素上の、少なくとも1つの入力機能ブロックおよび少なくとも1つの出力機能ブロックが、共通フィールドバスセグメント上のフィールドバスフィールド装置に割り当てられるかどうかを判定してもよい。入力/出力ブロックが、共通フィールドバスセグメント上のフィールドバスフィールド装置に適切に割り当てられない場合、ルーチン42は、フィールドバスセグメントへの制御モジュールの完全な同期割り当ては可能ではないと判定し、したがって、そのモジュールのいずれの機能ブロックも、フィールドバスセグメント上のフィールドバスI/O装置のフィールドバスポートには割り当てない。この場合、ルーチン42はブロック67に制御を与え、ブロック67では、フィールドバスI/O装置への制御モジュールの自動割り当ては失敗したことをユーザに通知する。ブロック64は、次に、未割り当ての制御機

10

20

30

40

50

能ブロックおよび入力/出力機能ブロックを、(図2のコントローラ57などの)フィールドバスセグメントに関連付けられたコントローラに割り当て、そして、ブロック61に制御を戻す。

【0039】

他方、ブロック66で、モジュールの入力/出力機能ブロックが共通フィールドバスセグメント上のフィールドバスフィールド装置に適切に割り当てられると判定した場合、次に、ブロック68で、フィールドバスセグメント上のフィールドバスI/O装置のフィールドバスポートの自動計算(auto-calculate)プロパティが「オン」(このI/O装置がフィールドバスセグメント上で機能ブロックを動作させる、または実行するように構成されていることを意味する)に設定されているかどうかを判定する。「オン」に設定されていない場合、ブロック67で、割り当ての失敗をユーザに通知し、そして、制御モジュール内の機能ブロックのコントローラへの従来の割り当てを行うブロック64に制御を与える。

10

【0040】

しかし、ブロック68でフィールドバスポートの自動計算プロパティがオンであると判定した場合は、ブロック69で、制御モジュールのクリティカルフロー経路内の未割り当ての機能ブロックのそれぞれをフィールドバスポートがサポートするかどうかを判定する。クリティカルフロー経路内の未割り当ての機能ブロックのうちの一つでもフィールドバスポートによってサポートされない場合、制御はブロック67および64に与えられ、ブロック67および64では割り当ての失敗をユーザに通知し、そして、未割り当ての機能ブロックのコントローラへの従来の割り当てを実行する。

20

【0041】

他方、フィールドバスポートがクリティカルフロー経路内の未割り当ての機能ブロックのそれぞれをサポートする場合、ブロック70で、制御モジュールのクリティカルフロー経路内の未割り当ての機能ブロックのそれぞれを実行するための十分なリソース(容量)をこのポートが有するかどうかを判定する。有さない場合、制御はブロック67および64に与えられ、ブロック67および64では、割り当ての失敗をユーザに通知し、そして、未割り当ての機能ブロックのコントローラへの従来の割り当てを実行する。他方、フィールドバスポートが容量を有する場合、ブロック71で、クリティカルフロー経路内の未割り当ての機能ブロックのそれぞれを、実行のためにフィールドバスI/O装置のフィールドバスポートに割り当てる。この時点で、所望される場合、ブロック71は、さらに、制御モジュール内の任意の残りの機能ブロック(すなわち、制御モジュールのクリティカルフロー経路内にないもの)がフィールドバスI/O装置のフィールドバスポートに割り当てられることが可能かどうか、または割り当てられるべきかどうかを判定してもよい。場合によっては、それらのブロックのすべてがフィールドバスI/O装置内で動作することが可能ならば、フィールドバスセグメント上で完全に同期した方法で実行される制御モジュールを提供するために、ブロック71でそれらのブロックをその装置に割り当ててもよい。あるいは、セグメント上で動作するその他の制御モジュールのためにこの装置のリソースを節約するために、ブロック71は、残りの機能ブロックのいずれもI/O装置に割り当てないように動作してもよい。所望される場合、ブロック71は、ユーザが選択可能な設定に基づいていずれかの方法で動作してもよい。いずれの場合も、現在の制御モジュールの機能ブロックの割り当てを完了した後は、制御は、その他の制御モジュールの処理のためにブロック61に戻される。

30

40

【0042】

もちろん、図3のフローチャート60は単に例示的なものであり、ルーチン42は、その他の方法で同様に動作してもよい。さらに、所望される場合、フローチャート60内のブロックのうちの特定のものは順番を並べ変えられてもよい。例えば、ブロック68、69、および70は、ルーチン60内のより先に、またはより後に配置されてもよく、順番を並べ変えられてもよく、そして、場合によっては制御モジュールのクリティカルフロー経路の決定に先立って実施されてもよい。

50

【 0 0 4 3 】

別の例では、自動ブロック割り当てのためのアルゴリズムは、以下のように要約される

。

【 0 0 4 4 】

【 数 1 】

Find synchronized control path

synchronizeList = new List

Foreach fieldbus input block in module

blockList = block. BlockFindSynchronisedControlPath()

append blockList to synchronizeList

End

loopExecutionIsSynchronised = false // Flag on module indicating possible

synchronized execution

loopExecutionOccursInH1Card = false // Flag on module indicating possible

execution of blocks in HI card

For each block in synchroniseList

loopExecutionIsSynchronised = true

If (block is not a Fieldbus block)

Mark block as potentially executing in the H1 card

loopExecutionOccursInH1Card = true

Endif

End

End

10

20

30

## 【数 2】

```

BlockFindSynchronisedControlPath()
    synchronizeList = new List
    If (thisBlock is a fieldbus output block)
        Append thisBlock to the synchroniseList
    ElseIf (thisBlock can be assigned to the HI port OR thisBlock is a Fieldbus
        block)
        Foreach output connector on thisBlock
            For each wire on the connector
                If the wire is not a feedback wire AND
                    the wire connects values of the exactly the same data type
                    AND the source connector is a predefined connector AND
                    the destination connector is a predefined connector
                        blockList = destinationBlock. BlockFind
                        SynchronisedControlPath()
                        If (blockList is not empty)
                            append blockList to synchroniseList
                            append thisBlock to synchroniseList
                        Endif
                    Endif
                Endif
            End
        End
    Endif
    Return synchroniseList
End

```

## 【0045】

図4～図9は、例示的な制御モジュールを示し、そして、ルーチン42が、上述の原理に基づいてそれらのモジュールの制御機能ブロックをコントローラに、またはフィールドバスセグメントに接続されたフィールドバスI/O装置のフィールドバスポートに割り当てる方法を示す。特に、図4は、PI機能ブロック77にリンクされたAI機能ブロック76を有し、PI機能ブロック77はフォワードおよびフィードバック方向の両方でAO機能ブロック78にリンクされている制御モジュール75を示す。公知でありかつ一般的であるように、AI機能ブロック76は測定信号またはその他の入力信号を取得し、そしてこの信号を示されているリンクを介してPI機能ブロック77に提供する。PI機能ブロック77は、比例/積分制御計算を実行して制御信号を生成する制御機能ブロックであり、その制御信号はAO機能ブロック78に提供される。AO機能ブロック78は、次に、制御信号を使用してバルブを開けること、または閉じること、あるいは、プラント内の物理的装置内の何らかのその他の設定を変更することなどの、プラント内の物理的機能を実行するように動作して制御動作を実行する。AO機能ブロック78は、さらに、例えば

現在のバルブ位置のフィードバック測定値を、P I 機能ブロック 77 による制御信号の作成における使用のために P I 機能ブロック 77 に提供する。

【 0 0 4 6 】

図 4 のモジュール 75 は、A I 機能ブロック 76 がフィールドバstransミッタ装置などのフィールドバス測定装置の中で動作するように、またはその中に配置されるようにスケジュールされる従来のフィールドバス測定ループ内にセットアップされる。このスケジュールリングは、A I 機能ブロック 76 の上の「 F F 」によって示されている。しかし、A O 機能ブロック 78 は、フィールドバス装置にはまったくスケジュールも割り当てもされないか、あるいは少なくとも、A I 機能ブロック 76 と同じフィールドバスセグメント内のフィールドバス装置にはスケジュールも割り当てもされない。結果として、ルーチン 42 は、自動割り当てを実行する際に、P I 機能ブロック 77 と A O 機能ブロック 78 との両方をフィールドバスセグメントに関連付けられたコントローラ 57 内に配置し、その理由は、モジュール 75 のクリティカルフロー経路内のすべての機能ブロックが、フィールドバスセグメント 56 上のフィールドバス I / O カード 58 と F F フィールド装置との中に配置されることはできないからである。この割り当て機能は、図 4 内の矢印によって示されている。

10

【 0 0 4 7 】

図 5 に示す制御モジュール 75 は、図 4 に示したものと同じであるが、ただし、従来のフィールドバスアクチュエータループ内にセットアップされる。ここでは、A O 機能ブロック 78 は、(このブロックの上の「 F F 」によって示されるように)フィールドバスセグメント上のフィールドバス装置(例えば、アクチュエータ)内に配置され、しかし、A I 機能ブロック 76 は、A O 機能ブロック 78 と同じフィールドバスセグメント上のフィールドバス装置には関連付けられない。ここでもやはり、同じフィールドバスセグメント上のフィールドバス装置に制御モジュール 75 のクリティカルフロー経路内の入力ブロックと出力ブロックとの両方は割り当てられないため(この場合は、入力ブロックが割り当てられないため)、ルーチン 42 は、A I 機能ブロック 76 と P I 機能ブロック 77 との両方をコントローラ 57 に割り当てる。

20

【 0 0 4 8 】

しかし、図 6 は、共通フィールドバスセグメント上のフィールドバス装置によって測定およびアクチュエーションの両方が実行される構成内にセットアップされる、制御モジュール 75 を示す。この動作は、入力機能ブロックおよび出力機能ブロック 76 および 78 のそれぞれの上の「 F F 」によって示されている。ここでは、制御モジュール 75 の入力機能ブロックおよび出力機能ブロックのそれぞれが、共通フィールドバスセグメント上のフィールドバス装置に割り当てられるため、ルーチン 42 は、P I 機能ブロック 77 を、フィールドバスセグメント 56 内での同期実行のためにフィールドバス I / O 装置 58 に(そして、より詳細には、その装置のフィールドバスポート 59 に)割り当てる。

30

【 0 0 4 9 】

図 7 は、カスケードタイプの制御ループを実施する別の制御モジュール 80 を示す。図 7 に示すように、制御モジュール 80 は、P I 制御機能ブロック 82 に接続された A I 機能ブロック 81 を有する第 1 ステージを含む。P I 制御機能ブロック 82 の出力は、第 2 ステージの P I 機能ブロック 83 への第 1 の入力として、第 2 ステージの A I 機能ブロック 84 からの追加の入力とともに提供される。第 2 ステージの P I 機能ブロック 83 は、2 つの入力をフィードバック信号とともに使用して制御信号を生成し、この制御信号は A O 機能ブロック 85 に提供される。

40

【 0 0 5 0 】

図 7 における「 F F 」の配置によって示されるように、A I 機能ブロック 81 および 84 と、A O 機能ブロック 85 とのそれぞれは、共通フィールドバスセグメント 56 上のフィールドバス装置に割り当てられる。結果として、ルーチン 42 は、フィールドバスセグメント 56 上での制御モジュール 80 の同期実行を提供するために、P I 機能ブロック 82 および 83 を、実行のためにフィールドバス I / O 装置 58 に割り当てるように動作す

50

る。

【 0 0 5 1 】

図 8 は図 7 のモジュール 8 0 を示しているが、この場合は、第 2 ステージの A I 機能ブロック 8 4 が、フィールドバスセグメント 5 6 上のフィールドバス装置に割り当てられないという点異なる。モジュール 8 0 のクリティカルフロー経路は、カスケードループの両方のステージの入力と出力とを含むため、そして、クリティカルフロー経路内の入力ブロックのうちの 1 つはモジュール 8 0 のその他の入力および出力ブロックと同じフィールドバスセグメント上のフィールドバス装置には割り当てられないかまたは関連付けられないので、ルーチン 4 2 は、クリティカルフロー経路内の未割り当てのブロックのすべて（すなわち、A I ブロック 8 4、ならびに P I 制御ブロック 8 2 および 8 3）をコントローラ 5 7 に割り当てる。なぜなら、P I ブロック 8 2 および 8 3 がフィールドバス I / O 装置 5 8 に割り当てられたとしても、制御モジュール 8 0 の同期実行は不可能だからである。

10

【 0 0 5 2 】

図 9 は、スプリットレンジ出力構成を有する別の制御モジュール 8 6 を示す。制御モジュール 8 6 は、P I 制御ブロック 8 8 に結合されたアナログ入力ブロック 8 7 を含み、P I 制御ブロック 8 8 は、スプリットレンジブロック ( S P L ) 8 9 に出力を提供し、スプリットレンジブロック ( S P L ) 8 9 は、さらに、2 つの並列接続された A O 機能ブロック 9 0 A および 9 0 B に結合される。ここでは、A I 機能ブロック 8 7 と、A O 機能ブロック 9 0 A および 9 0 B とのそれぞれが、共通フィールドバスセグメント上のフィールドバス装置に割り当てられるので、ルーチン 4 2 は、フィールドバスセグメント 5 6 上でのモジュール 8 6 の同期実行を提供するために、P I 機能ブロック 8 8 と S P L 機能ブロック 8 9 とを実行のためにフィールドバス I / O 装置 5 8 に割り当てる。この動作は、もちろん、装置 5 8（そして、より詳細には、この装置のフィールドバスポート 5 9）が、P I および S P L ブロックの機能をサポートし、かつ、これらのブロックの両方を実行するための十分なリソースを有するというを前提としている。しかし、この例は、モジュールの入力機能ブロックと出力機能ブロックとの間の、モジュールのクリティカルフロー経路内で直列に接続された複数のブロック（例えば、制御ブロック）が、フィールドバス I / O 装置に割り当てられることが可能であることを示している。

20

【 0 0 5 3 】

いずれの場合も、ルーチン 4 2（図 1）が、上述のように、作成された制御モジュールのそれぞれの機能ブロックをフィールドバスセグメントに関連付けられた装置に割り当てるように動作した後は、ルーチン 4 4 が、ルーチン 4 2 によって作成された割り当てに基づいて、フィールドバスセグメントのための通信および実行スケジュールを自動的に生成するために使用されてもよい。一般的に言えば、ルーチン 4 4 は、フィールドバスセグメントのための通信および実行スケジュールの作成をそのセグメント上で動作する機能ブロックのそれぞれを識別することと、そのセグメントためのマクロサイクル（マクロサイクル内の機能ブロックのそれぞれについての通信および実行のためのオフセット時間を指定する）を作成することとによって行う。もちろん、ルーチン 4 4 は、複数の異なる制御モジュールからの機能ブロックを含むマクロサイクルを作成してもよい。

30

40

【 0 0 5 4 】

理解されるように、ルーチン 4 0 の使用は、ユーザインタフェース装置における、制御の定義および表示のための統一された環境を提供する。特に、ルーチン 4 0 を使用すれば、制御ブロックが実行のためにどこに割り当てられるか（すなわち、コントローラ内か、フィールドバスインタフェースカード、または、特定のフィールドバスセグメント上のフィールドバス装置内か）を考慮することなく、制御を定義することが可能である。この特徴が可能である理由は、コントローラ内で動作可能な機能ブロックのほとんどは、フィールドバスインタフェースカード内で実行することも許可されるからである。したがって、ユーザは、測定または作動のために選択するフィールドバス装置が、制御または監視アプリケーションを実行するために必要とされる機能ブロックをサポートするかどうかを心配

50

する必要はなく、なぜなら、フィールドバスフィールド装置内でサポートされない任意のブロックはフィールドバスインタフェースカードに割り当てられてもよく、そして、それでもフィールドバスセグメント上で同期して動作することが可能であるからである。

#### 【 0 0 5 5 】

一実施形態では、ルーチン 4 4 は、フィールドバスセグメント上で使用されるスケジュールを、このセグメントへの機能ブロックの割り当てに基づいて自動的に計算してもよく、そして、そうすることにより、フィールドバスセグメント上での複数の異なる実行レートの機能ブロックのサポートを自動的に提供してもよい。例えば、複数の制御ループ（またはモジュール）が同じフィールドバスセグメント上で実行される場合、ループのそれぞれにとって同じ実行レートは適切ではない可能性がある。同様に、場合によっては、1つの制御ループ内に、異なるレートで実行される機能ブロックを有することが望ましい可能性がある。フィールドバスセグメント上の、いくつかの制御モジュールの、または制御モジュール内のいくつかの機能ブロックの、相対的な実行レートをユーザが選択的に変更することを可能にするために、ユーザは異なる実行レートを有するモジュール（または機能ブロック）を構成してもよい。あるいは、同じセグメント上の異なる機能ブロックについての相対的な実行レートを指定するために、実行マルチプライヤが、フィールドバスポート内で、またはフィールドバス装置内で実行されるように割り当てられるブロックに提供されてもよい。

#### 【 0 0 5 6 】

例として、図 1 0 は、P I D 制御機能ブロック 9 3（P I D 2 とラベル付けされている）に接続された A I 機能ブロック 9 2（A I 2 とラベル付けされている）を有する第 1 ステージを含む、カスケード制御ループを実施する制御モジュール 9 1 を示す。P I D 制御機能ブロック 9 3 の出力は、第 2 ステージの P I D 機能ブロック 9 4（P I D 1 とラベル付けされている）への第 1 の入力として、第 2 ステージの A I 機能ブロック 9 5（A I 1 とラベル付けされている）からの追加の入力とともに提供される。第 2 ステージの P I D 機能ブロック 9 4 は、2 つの入力を、フィードバック信号とともに使用して制御信号を生成し、この制御信号は A O 機能ブロック 9 6 に提供される。図 1 0 に示すように、入力および出力機能ブロック 9 2、9 5、および 9 6 は、同じセグメント 5 6 上のフィールドバス装置に割り当てられるので、ルーチン 4 2 は、セグメント 5 6 上でのモジュール 9 1 の同期実行を提供するために、P I D 機能ブロック 9 3 および 9 4 をフィールドバス I / O 装置 5 8 のフィールドバスポート 5 9 に割り当てる。

#### 【 0 0 5 7 】

しかし、図 1 0 に示すように、A I 機能ブロック 9 2 と P I D 機能ブロック 9 3 とは、それらに関連付けられた 4 x マルチプライヤを有する（それらのマルチプライヤは、構成プロセスの間に、構成エンジニアまたはその他のユーザによって指定されてもよい）。それらのブロックについての、4 x マルチプライヤの指示は、それらのブロックが、制御モジュール 9 1 内のその他のブロックのレートよりも 4 倍遅いレートで動作すべき、または実行されるべきであるということ指定する。特に、カスケードループの 1 次（第 1）ステージは、そのループの 2 次ステージよりも 4 倍遅く実行される必要があるため、1 次ステージの A I および P I D ブロック 9 2 および 9 3 は、4 x 実行マルチプライヤを使用して構成される。ルーチン 4 4 は、フィールドバスセグメント 5 6 のためのスケジュールを生成する際に、カスケードループの A I および P I D ブロック 9 2 および 9 3 が、その他の機能ブロックよりも 4 倍遅く実行されるべきであることを考慮に入れ、そして、そうすることにより、ルーチン 4 4 は、実行レートにおけるこの変化を含むマクロサイクル（すなわち、機能ブロック実行スケジュール）を作成する。特に、フィールドバスセグメントのための 1 つのマクロサイクル内で、（4 x マルチプライヤを割り当てられた）機能ブロック 9 2 および 9 3 は、1 回だけスケジュールされ、一方、その他の機能ブロックのすべては、マクロサイクル内で 4 回実行されるように割り当てられる。

#### 【 0 0 5 8 】

図 1 1 は、セグメント 5 6 が図 1 0 の制御モジュール 9 1 のみを実行する場合に、セグ

10

20

30

40

50

メント56のためにルーチン44によって生成される、例示的な、結果として得られるマクロサイクル97を示す。ここでは、マクロサイクル97は、4つの等しい時間周期に分割されていることがわかるであろう(時間周期のうちの1つは、「繰り返す時間(time to repeat)」の矢印によって示されている)。2次ステージのAIブロック95、PIDブロック94、およびAOブロック96のそれぞれは、繰り返す時間のそれぞれの間に実行されるように(すなわち、マクロサイクル97の間に4回実行されるように)スケジュールされ、一方、AIブロック92およびPIDブロック93は、マクロサイクル97の間に1回だけ実行されるようにスケジュールされる。したがって、1次ステージのAIおよびPIDブロック92および93は、2次ステージのブロック94~96よりも4倍遅いレートで実行される。

10

## 【0059】

モジュールの2つのブロックが異なるレートで実行される場合は、通常、それらの実行を相互に同期させる必要はなく、したがって、異なるレートで動作するブロックの実行時間は、それらのブロックがフィールドバスセグメント上の異なる装置内で実行される限り、一般にオーバーラップしてもよい。多くの場合、この特徴は、図11の例示的なマクロサイクル(ここで、繰り返す時間は、制御モジュール91の2次ステージを実行するために必要とされる最小時間である)によって示されるように、より遅く実行されるブロックに関連付けられた通信が、より速く実行されるブロックに時間遅延を本質的に追加しないような方法でスケジュールされることを可能にする。

## 【0060】

20

所望される場合、異なる実行レートを有する機能ブロックを同じフィールドバスセグメント上に含むために繰り返す時間を決定し、そして次に、フィールドバスセグメントのマクロサイクルを決定することにおいて、ルーチン44によって以下の手順が使用されてもよい。最初に、ルーチン44は、実行マルチプライヤを割り当てられていない機能ブロック(すなわち、最も速い実行レートを有する機能ブロック)の実行レートと、非周期の通信をサポートするために追加される時間とに基づいて、繰り返す時間(または繰り返し時間(repeat time))を計算してもよい。これは、マクロサイクルの、提案される繰り返し時間となる。ルーチン44は、次に、各繰り返し時間の間の、それらのブロック(すなわち、最も速く動作するブロック)の1回の実行についての実行時間を、それらのブロックの、それぞれの制御モジュール内の信号フロー順序に基づいて順序付けしてもよく、そして、可能な場合は、機能ブロックの並列実行を最大限に利用してもよい。次に、ルーチン44は、マクロサイクルを、マクロサイクル=繰り返し時間×設定されている最大の実行マルチプライヤ、として計算してもよい。

30

## 【0061】

次に、ルーチン44は、残りのブロック(すなわち、実行マルチプライヤを有するブロック)に関連付けられた通信を、マクロサイクル全体にわたる利用可能な時間内で、層状に重ねてもよい。ルーチン44がその他のブロックのすべての通信をマクロサイクル内に収めることができない場合、ルーチン44は、より多くの通信がマクロサイクル内でスケジュールされることを可能にするために、(例えば、20パーセントだけ)繰り返し時間を拡張してもよい。ルーチン44は、次に、より遅い実行レートを有するブロックのための通信を新しいマクロサイクル内に含めることと、必要ならば繰り返し時間を拡張することとのプロセスを、セグメント上のすべての機能ブロックのためのすべての通信が、結果として得られるマクロサイクル内でスケジュールされることが可能になるまで繰り返してもよい。

40

## 【0062】

図12は、複数のモジュールが同じフィールドバスセグメント上での実行のためにスケジュールされた、別のマクロサイクル98を示す。この例では、4つの単純なPID制御モジュール(それぞれが、1つのAI、AO、およびPID機能ブロックを有する)が、フィールドバスセグメント上で構成されている。ここでは、各制御モジュールのAIおよびAOブロックは、セグメント上のフィールドバスフィールド装置内に配置され、そして

50

、PIDブロックは、フィールドバスセグメントのフィールドバスI/O装置のフィールドバスポートに自動的に割り当てられると仮定されている。しかし、モジュールのうちの1つ(AI1、PID1、およびAO1ブロックを含むもの)は、例えば液体圧を制御してもよく、そして、その他の制御モジュールと比較してできるだけ速く実行する必要がある。その他のモジュールは、この例ではそれほどクリティカルではなく、そして、より速いモジュールの実行レートよりも4倍遅いレートで実行されてもよい。上記の手順を使用して計算された、セグメントのための通信および実行スケジュールのマクロサイクル98を図12に示す。ここで、(AI1、PID1、およびAO1機能ブロックを有する)第1のモジュールは、マクロサイクル98の間に4回実行され、一方、その他のモジュールは、マクロサイクル98の間に1回だけ実行される。この特徴を実施するためには、第1のモジュールが、(このモジュールはその他のモジュールよりも4倍速く動作すべきであるという事実により)4つの繰り返し時間のそれぞれの間での実行のためにスケジュールされ、そしてその他のモジュールが、次に、結果として得られるマクロサイクル98内の残りの時間の中でスケジュールされる。

#### 【0063】

一般的に言えば、ルーチン44は、フィールドバスセグメントのための2つのスケジューリング技術のうちの1つをユーザが選択することを可能にするように動作してもよく、指定されるスケジューリング技術は、フィールドバスポート上の最適自動計算(optimal auto-calculate)プロパティを設定することによって選択される。2つのスケジューリング技術は、マクロサイクルごとに1回のブロック実行を許可する従来のスケジューリングと、サブスケジュールを有する1つのマクロスケジュールをサポートする最適自動計算スケジュールマクロサイクルとを含む。最も速いサブスケジュールの周期は、最も速いモジュール実行時間と機能ブロックマルチプライヤとの組み合わせによって設定される。この場合、各サブスケジュールは、その独自の周期を有してもよい。さらに、マクロサイクルは、ポート(レガシー)のために設定されたモジュール実行時間に、または、設定された時間が達成可能ではない場合は最小の時間に基づく。より遅く実行されるいくつかのブロックをサポートするサブスケジュールは、設定された実行マルチプライヤとポートマクロサイクル時間とに基づく。

#### 【0064】

所望される場合、モジュール実行時間は、マルチプライヤを使用して特定のフィールドバスセグメント時間内にマッピングされてもよい。一般に、基準セグメント時間(base segment time)は、100、125、128、またはその倍数などの、任意の数であってもよい。制約は、基本的に、サブスケジュール時間は2の倍数でなければならないということと、サポートされる最も遅いサブスケジュールは最も速いサブスケジュールよりも最大で8倍遅いということとであるが、それらの制約の代わりに、またはそれらの制約に加えて、その他の制約が使用されてもよい。所望される場合、自動計算スケジュール特徴は、1、2、4、および8の実行マルチプライヤをサポートする。ここで、4x遅いという指示は、マクロスケジュールが4つのセグメントに分割されるということの意味し、8x遅いという指示は、マクロスケジュールが8つのセグメントに分割されるということの意味し、以下同様である。

#### 【0065】

もちろん、2つよりも多くの実行レートがサポートされてもよい。この場合、最も遅く実行されるブロックによって使用されるマルチプライヤに基づいた回数だけ繰り返される繰り返し時間を有する、最も速い実行レートが最初にスケジュールされる。したがって、マルチプライヤを有さないブロックと、2x、4x、および8xマルチプライヤを有するブロックとがある場合、(マルチプライヤを有さない)最も速く実行されるブロックについてのサブスケジュールは、マクロサイクルの間に8回繰り返される。次に最も速い実行レートを有するブロック(例えば、2xブロック)が、次に、第1のサブスケジュールを使用して作成されたマクロサイクル内の、残りに時間にわたってスケジュールされる。この第2のサブスケジュールの繰り返し時間は、それらのブロックのマルチプライヤの、最

10

20

30

40

50

も遅く実行されるブロックのマルチプライヤに対する比（すなわち、この場合は  $2 \times / 8 \times$  であり、4 に等しい）に基づく。これらのブロックが、第1のサブスケジュールを使用して作成されたマクロサイクル内でスケジュールされることができない場合、第1のサブスケジュールの繰り返し時間は増加させられ、そしてこのプロセスは、次に最も速い実行ブロックが、結果として得られるマクロサイクル内でスケジュールされることが可能になるまで繰り返される。このプロセスは、次に、異なるレート（またはサブスケジュール）のそれぞれについて、すべてのブロックが、結果として得られるマクロサイクル内でスケジュールされることが可能になるまで繰り返される。

【0066】

もちろん、モジュール走査レートと機能ブロックマルチプライヤとの両方が、スケジュール内で使用される。例えばカスケードループ内において、機能ブロックマルチプライヤを使用することが重要である場合が存在する。マルチプライヤは、モジュールの相対実行レートによって決定される。マクロスケジュールおよびサブスケジュールの決定において、マクロスケジュールは最も速いループによって設定され、次に、マルチプライヤ（最大  $8 \times$ ）が、その他のモジュールのために使用される。サブスケジュールの実行レートを達成できない場合は、その状況がユーザに警告されてもよい。所望される場合は、しかし、それでもスケジュールはダウンロードされる。ユーザは、セグメント実行レートを無効にすることはできない（例外としては、もちろん、ユーザはスケジュール全体を無効にすることができる）。

【0067】

一般に、フィールドバスセグメント上で複数の異なるブロック実行レートを構成およびサポートする1つの方法について、図13～図35に関してより詳細に説明する。特に、上述のように、プロセス制御システム内のプロセスコントローラとフィールド装置とによって機能ブロックを実行するためのスケジュールを生成するために使用される公知の技術は、通常、同じ通信バス（例えば、機能ブロックを実行するために割り当てられたフィールド装置とコントローラとを通信可能に結合する通信バス）に関連付けられたすべての機能ブロックを、最も遅く実行される機能ブロックのレートで実行するようにスケジュールすることを含む。したがって、公知の技術では、相対的により速く実行される機能ブロックのブロック実行周期を、最も遅く実行される機能ブロックのブロック実行周期に一致するように増加させること（例えば、機能ブロック実行レートを遅くすること）が必要とされ、したがって、一部の機能ブロックのより短い実行周期（すなわち、より速い実行レート）が、プロセス制御システムを実施するために有利に使用されることはできない。

【0068】

機能ブロックを実行するためのスケジュールを実施するために使用されるいくつかの公知の技術とは異なり、以下で述べる例示的な方法および装置は、機能ブロックを実行するためのスケジュールを、それらのそれぞれの機能ブロック実行周期に基づいて、すなわち、同じ通信バスに関連付けられた（例えば、同じ通信バスを介して通信する）その他の機能ブロックの機能ブロック実行周期に一致するように機能ブロック実行周期を実質的に増加させる必要なしに、生成するために使用されてもよい。いくつかの機能ブロックを、同じ通信バスに関連付けられた、その他のより遅い機能ブロックよりも速く実行することは有利な場合がある。例えば、圧力測定機能ブロックとバルブ制御機能ブロックとが同じ通信バスに関連付けられている場合、圧力測定機能ブロックをより短い周期を使用して（例えば、より速いレートで）実行することは有利な可能性があり、一方、バルブ制御機能ブロックを、より短い周期を使用して実行することには、あまり利点はない可能性がある。したがって、圧力測定機能ブロックの機能ブロック実行周期は、バルブ制御機能ブロックの機能ブロック実行周期よりも短くてもよい。機能ブロックの実行をスケジュールするための公知の技術を使用する場合、圧力測定機能ブロックが、バルブ制御機能ブロックと同じ、より遅い周期を使用して実行されることが要求される。結果として、圧力測定機能ブロックは、プロセス内のより高い頻度の圧力変化を捕捉し損なう可能性がある。対照的に、本明細書に記載された例示的な方法および装置を使用して、機能ブロックを実行するた

10

20

30

40

50

めのスケジュールをそれらのそれぞれのブロック実行レートを使用して生成する場合、圧力測定機能ブロックがバルブ制御機能ブロックよりもより短い周期を使用して（例えば、より速い実行レートで）実行されることが可能となる。したがって、圧力測定機能ブロックは、圧力測定値をより頻繁に（例えば、相対的に高い分解能で）取得して、例えば、公知の機能ブロックスケジューリング技術を使用する場合には捕捉も処理もされないであろうより高い頻度の圧力変化（例えば、ブリップ、スパイク、またはその他の比較的高い頻度の圧力変化）を捕捉することが可能である。

【 0 0 6 9 】

本明細書に記載されているように、それぞれの機能ブロックのブロック実行レートを使用してスケジュールを生成することは、複数のプロセスループを、同じ通信バスに通信可能に結合されたフィールド装置またはコントローラによって実行されるようにスケジュールすることと、さらに、プロセスループがそれらのそれぞれのループ実行周期で実行されてもよいことを保証することとを可能にする。すなわち、より短いループ実行周期（例えば、より速いループ実行レート）に関連付けられたプロセスループは、より長いループ実行周期（例えば、より遅いループ実行レート）を有するプロセスループよりも相対的に速く実行されてもよい。このように、機能ブロックを実行するためのスケジュールを生成するために使用されるいくつかの公知の方法とは異なり、同じ通信バスに関連付けられたすべてのループのループ実行周期は、最も長いループ実行と同じにされる必要はない。

【 0 0 7 0 】

次に、図 1 3 を参照すると、本明細書に記載された例示的な方法および装置を実施するために使用され得る例示的なプロセス制御システム 1 0 0 は、ワークステーション 1 0 2（例えば、アプリケーションステーション、オペレータステーションなど）と、コントローラ 1 0 6 とを含み、それらの両方は、アプリケーション制御ネットワーク（ACN）と一般に呼ばれるバスまたはローカルエリアネットワーク（LAN）1 0 8 を介して通信可能に結合されてもよい。LAN 1 0 8 は、任意の所望の通信媒体およびプロトコルを使用して実施されてもよい。例えば、LAN 1 0 8 は、ハードワイヤードまたはワイヤレスのイーサネット（登録商標）通信方式に基づいてもよく、この通信方式は周知であり、したがって、本明細書ではさらに詳細には説明しない。しかし、当業者によって容易に理解されるように、任意のその他の好適な通信媒体およびプロトコルが使用されてもよい。さらに、1 つの LAN が示されているが、2 つ以上の LAN と、ワークステーション 1 0 2 内の適切な通信ハードウェアとが、ワークステーション 1 0 2 とそれぞれの類似したワークステーション（図示せず）との間の冗長通信経路を提供するために使用されてもよい。

【 0 0 7 1 】

ワークステーション 1 0 2 は、1 つ以上の情報技術アプリケーション、ユーザ対話型アプリケーション、および/または通信アプリケーションに関連付けられた動作を実行するように構成されてもよい。例えば、ワークステーション 1 0 2 は、プロセス制御関連アプリケーションに関連付けられた動作と、ワークステーション 1 0 2 およびコントローラ 1 0 6 が任意の所望の通信媒体（例えば、ワイヤレス、ハードワイヤードなど）およびプロトコル（例えば、HTTP、SOAP など）を使用してその他の装置またはシステムと通信することを可能にする、通信アプリケーションに関連付けられた動作とを実行するように構成されてもよい。ワークステーション 1 0 2 は、任意の好適なコンピュータシステムまたは処理システム（例えば、図 3 5 のプロセッサシステム 2 3 1 0）を使用して実施されてもよい。例えば、ワークステーション 1 0 2 は、シングルプロセッサのパーソナルコンピュータ、シングルプロセッサまたはマルチプロセッサのワークステーションなどを使用して実施されてもよい。

【 0 0 7 2 】

コントローラ 1 0 6 は、システムエンジニアまたはその他のシステムオペレータによって、ワークステーション 1 0 2 または任意のワークステーションを使用して生成された、かつ、コントローラ 1 0 6 にダウンロードされ、そしてコントローラ 1 0 6 内でインスタンス化された、1 つ以上のプロセス制御ルーチンを実行してもよい。コントローラ 1 0 6

は、例えば、フィッシャーローズマウントシステムズ社 ( Fisher - Rosemount Systems, Inc. ) とエマーソンプロセスマネジメント ( Emerson Process Management ( 登録商標 ) ) とによって販売されている、デルタV ( DeltaV ( 登録商標 ) ) コントローラであってもよい。しかし、任意のその他のコントローラが代わりに使用されてもよい。さらに、図13には1つのみのコントローラが示されているが、任意の所望のタイプの、または任意の所望のタイプを組み合わせた追加のコントローラが、LAN108に結合されてもよい。

【0073】

コントローラ106は、デジタルデータバス114と、入出力(I/O)装置116とを介して、複数のフィールド装置112a~cに結合されてもよい。プロセス制御ルーチンの実行中に、コントローラ106は、フィールド装置112a~cと情報(例えば、コマンド、構成情報、測定情報、ステータス情報など)を交換してもよい。例えば、コントローラ106は、コントローラ106によって実行された場合に、コントローラ106がフィールド装置112a~cにコマンドを送信することをもたらずプロセス制御ルーチンを備えてもよく、それらのコマンドは、フィールド装置112a~cが指定された動作(例えば、測定の実行、バルブの開閉など)を実行すること、かつ/または、デジタルデータバス114を介して情報(例えば、測定データ)を通信することをもたらずのものであってもよい。

【0074】

プロセス制御システム100のフィールド装置112a~cを識別するために、フィールド装置112a~cのそれぞれは、固有の物理的装置タグ(PDT)を備えてもよい(例えば、記憶してもよい)。例えば、第1のフィールド装置112aの物理的装置タグはPDT1であり、第2のフィールド装置112bの物理的装置タグはPDT2であり、第3のフィールド装置112cの物理的装置タグはPDT3である。図示されている例では、フィールド装置112a~cは、第1の圧カトランスミッタ112aと、第2の圧カトランスミッタ112bと、デジタルバルブコントローラ(DVC)112cとを含む。しかし、任意のその他のタイプのフィールド装置(例えば、バルブ、アクチュエータ、センサなど)が、本明細書に記載された例示的な方法および装置に関連して使用されてもよい。

【0075】

図示されている例では、フィールド装置112a~cは、周知のフィールドバスプロトコルを使用してデジタルデータバス114を介して通信するように構成されたフィールドバス準拠の装置である。フィールドバス規格により、デジタルデータバス114は、測定および制御装置(例えば、フィールド装置112a~c)に通信可能に結合されるように構成された、デジタル、双方向、マルチドロップ通信バスである。フィールド装置112a~cは、マルチドロップ構成内でデジタルデータバス114に通信可能に結合されて示されている。1つのフィールド装置にI/O装置116と通信するためのデジタルデータバスの排他的使用が与えられるポイントツーポイント構成を使用して、フィールド装置をI/O装置116に通信可能に結合するための、デジタルデータバス114または類似したデータバスが代わりに使用されてもよい。代替の例示の実施では、フィールドバス準拠の装置を含んでもよく、または含まなくてもよい、その他のタイプのフィールド装置(例えば、周知のプロフィバス(Profibus)およびハート(HART)通信プロトコルを使用して、データバス114を介して通信する、プロフィバスまたはハート準拠の装置)に関連して、本方法および装置が使用されてもよい。デジタルデータバス114は、本明細書ではセグメントとも呼ばれる。セグメントは、その特性インピーダンスで終端される物理的バスを表すフィールドバス用語である。図13に示されている例では、デジタルデータバス114がセグメントを形成する。本明細書に記載された例示的な方法および装置は、1つのセグメント(例えば、デジタルデータバス114)に関連して、または、より長い論理的バスを形成するためにリピータを使用してリンクされた、2つ以上のセグメント(例えば、デジタルデータバス114と、1つ以上のその他のデジタルデータバス

10

20

30

40

50

)に関連して実施されてもよい。

【0076】

図示されている例では、I/O装置116は、コントローラ106とフィールド装置112a~cとを、フィールドバスプロトコル、またはその他のタイプの通信プロトコル(例えば、プロフィバス(Profibus)プロトコル、ハート(HART)プロトコルなど)を使用し得るその他のフィールド装置に接続することを可能にする、I/Oサブシステムインタフェースを使用して実施される。例えば、I/O装置116は、フィールドバスプロトコルと、その他の通信プロトコルとの間の変換を行う1つ以上のゲートウェイを含んでもよい。フィールド装置の追加のグループが、コントローラ106と通信することを可能にするために、(I/O装置116に類似した、またはI/O装置116と同一の)追加のI/O装置がコントローラ106に結合されてもよい。

10

【0077】

例示的なプロセス制御システム100は、以下でさらに詳細に説明する例示的な方法および装置が有利に使用されてもよい、1つのタイプのシステムを説明するために提供される。しかし、本明細書に記載された例示的な方法および装置は、所望される場合、図13に示す例示的なプロセス制御システム100より複雑な、またはより単純なその他のシステム内で、および/または、プロセス制御活動、企業管理活動、通信活動などに関連して使用されるシステム内で有利に使用されてもよい。

【0078】

図13に示されている例では、コントローラ106および/またはフィールド装置112a~cによって実行されるプロセス制御ルーチンを定義するためのモジュール120が、ワークステーション102において構成される。モジュール120は、プロセス制御ルーチンを実施するためにフィールド装置112a~cによって実行される機能を定義する複数の機能ブロック122a~eを含む。これら機能は、フィールド装置112a~cに、測定値(例えば、圧力値、温度値、フロー値、電圧値、電流値など)の取得、アルゴリズムまたは計算(例えば、積分、微分、加算、減算など)の実行、計装(例えば、バルブの開閉、炉の調節、ボイラの調節など)の制御を行わせてもよく、または任意のその他の機能を実行させてもよい。図示されている例では、フィールド装置112a~cは、機械実行可能命令の形態で、機能ブロック122a~eのうちのそれぞれによって定義される機能を記憶して実行する。しかし、他の例示的な実施では、モジュール120は、フィールド装置112a~cのうちの1つによってではなく、コントローラ106によって実行される機能を表す機能ブロックを、機能ブロック122a~eのうちの1つ以上の代わりに、またはそれらに加えて備えてもよい。

20

30

【0079】

ワークステーション102は、さらに、フィールド装置112a~cおよび/またはコントローラ106によって実行される、1つ以上のその他の機能ブロック(図示せず)を有する、別のモジュール124を構成するために使用されてもよい。2つのモジュール(モジュール120および124)が示されているが、コントローラ106および/またはフィールド装置112a~cによって実行される追加の機能ブロックを有するより多くのモジュールが、ワークステーション102において構成されてもよい。他のモジュール(1つまたは複数)は、他のプロセス制御ルーチンを実施するために、かつ/または、モジュール120および124に関連してプロセス制御ルーチンを実施するために使用されてもよい。

40

【0080】

図示されている例では、機能ブロック122a~eは、第1のアナログ入力(AI1)機能ブロック122aと、第1の比例/積分/微分(PID1)機能ブロック122bと、第2のアナログ入力(AI2)機能ブロック122cと、PID2機能ブロック122dと、アナログ出力(AO1)機能ブロック122eとを含む。AI1機能ブロック122aとPID1機能ブロック122bとは、フィールド装置112aによって実行される機能を定義する。AI2機能ブロック122cは、フィールド装置112bによって実行

50

される機能を定義する。PID 2 機能ブロック 1 2 2 d と AO 1 機能ブロック 1 2 2 e とは、フィールド装置 1 1 2 c によって実行される機能を定義する。代替の例示的实施では、機能ブロック 1 2 2 a ~ e の代わりに、またはそれらに加えて、任意のその他のタイプの機能ブロックが使用されてもよい。

#### 【 0 0 8 1 】

図 1 4 は、機能ブロック 1 2 2 a ~ e のグラフィカルユーザインタフェース (GUI) 表示を示す。機能ブロック 1 2 2 a ~ e は、例えばワークステーション 1 0 2 によって実行される、GUI ベースの設計ソフトウェアアプリケーションを使用して、ユーザ (例えば、エンジニア、オペレータなど) によって相互接続されてもよい。図 1 4 に示すように、機能ブロック 1 2 2 a ~ e のそれぞれは、1 つ以上の入力、および / または 1 つ以上の出力を含む。機能ブロック 1 2 2 a ~ e の入力および / または出力を接続することによって、モジュール 1 2 0 (図 1 3) のプロセス制御ルーチンは定義される。機能ブロック 1 2 2 a ~ e の間の接続は、本明細書では、機能ブロック接続線 (function block connection wires) と呼ばれる。図示されている例では、AI 1 機能ブロック 1 2 2 a の出力は、PID 1 機能ブロック 1 2 2 b の入力に接続され、PID 1 機能ブロック 1 2 2 b および AI 2 機能ブロック 1 2 2 c の出力は、PID 2 機能ブロック 1 2 2 d の入力に接続され、そして、PID 2 機能ブロック 1 2 2 d の出力は、AO 1 機能ブロック 1 2 2 e の入力に接続される。

#### 【 0 0 8 2 】

図 1 5 を簡単に参照すると、AI 2 機能ブロック 1 2 2 c から PID 2 機能ブロック 1 2 2 d に情報を通信することを可能にするために AI 2 機能ブロック 1 2 2 c の出力 2 0 4 が PID 2 機能ブロック 1 2 2 d の入力 2 0 6 に結合されてもよい方法を示す例示的目的のために、例示的な機能ブロック結合構成 2 0 2 (すなわち、結合構成 2 0 2) が提供されている。結合プロセスが、機能ブロック (例えば、図 1 3 および図 1 4 の機能ブロック 1 2 2 a ~ e) の間の接続に基づいて結合構成 2 0 2 を生成する。結合プロセスは、機能ブロック相互接続に従って機能ブロックが情報を交換することを可能にするために新しい機能ブロック接続 (例えば、出力 2 0 4 と入力 2 0 6 との間の接続) が作成されるときはいつでも、例えばワークステーション 1 0 2 によって実行されてもよい。

#### 【 0 0 8 3 】

結合プロセスは、装置内 (intra-device) 通信を可能にするための装置内リンクと、装置間 (inter-device) 通信を可能にするための装置間リンクとを作成するために構成される。装置内リンクは、同じ装置に関連付けられた 2 つの機能ブロックの間の接続を定義する。例えば、装置内リンクは、AI 1 機能ブロック 1 2 2 a (図 1 3 および図 1 4) と PID 1 機能ブロック 1 2 2 b (図 1 3 および図 1 4) との間の接続を定義し、なぜなら、機能ブロック 1 2 2 a および 1 2 2 b によって定義される機能は、同じ装置 (図 1 3 のフィールド装置 1 1 2 a) によって実行されるからである。装置間リンクは、1 つのフィールド装置内の機能ブロックと、別のフィールド装置内の機能ブロックとの間の接続 (2 つのフィールド装置を通信可能に結合している通信バス (例えば、図 1 3 のデジタルデータバス 1 1 4) を介してフィールド装置が通信することをもたらす) を定義する。例えば、装置間リンクは、AI 2 機能ブロック 1 2 2 c (図 1 3 ~ 図 1 5) と PID 2 機能ブロック 1 2 2 d との間の接続を定義し、なぜなら、AI 2 機能ブロック 1 2 2 c に関連付けられた機能は、フィールド装置 1 1 2 b (図 1 3) によって実行され、PID 2 機能ブロック 1 2 2 d に関連付けられた機能は、フィールド装置 1 1 2 c (図 1 3) によって実行されるからである。

#### 【 0 0 8 4 】

図 1 5 に示されている例では、結合プロセスは、AI 2 機能ブロック 1 2 2 c を PID 2 機能ブロック 1 2 2 d に結合するための装置間リンク構成を作成する。最初に、結合プロセスは、出力 2 0 4 を入力 2 0 6 にリンクするための情報を含む、装置 - 装置 (device-to-device) リンクオブジェクト 2 0 8 (例えば、装置間リンクオブジェクト) を作成する。同じ装置に関連付けられた機能ブロックを結合するために使用され

10

20

30

40

50

る代替の例示的实施では、結合プロセスは、装置内リンクオブジェクト（図示せず）を作成する。図15の例示的な結合構成202では、結合プロセスは、次に、発行者（publisher）リンク210を作成し、発行者リンク210を出力204と関連付け、そして発行者リンク210を装置-装置リンクオブジェクト208に結合する。さらに、結合プロセスは、引用者（subscriber）リンク212を作成し、引用者リンク212を入力206と関連付け、そして引用者リンク212を装置-装置リンクオブジェクト208に結合する。

#### 【0085】

結合プロセスは、さらに、発行者仮想通信路（virtual communication resource）（VCR）214と、引用者VCR216とを作成する。VCRは機能ブロック間の接続の識別情報を維持し（または存続させ）、それにより、機能ブロック間の任意の通信がVCR識別情報を使用して行われることが可能になる。図示されている例では、発行者VCR214は、AI2機能ブロック122cの固有の識別情報218を発行者リンク210と関連付け、そして、引用者VCR216は、PID2機能ブロック122dの固有の識別情報220を引用者リンク212と関連付ける。一般に、発行者/引用者VCR（例えば、発行者および引用者VCR214および216）は、（例えば、1つの機能ブロックが多数の機能ブロックに情報をブロードキャストする）1対多数同報通信を含む機能ブロック間のバッファリング付きの通信（buffered communications）を可能にする。図示されている例では、AI2機能ブロック122cは、新しいデータを生成した後で、そのデータを、出力204を介してPID2機能ブロック122dに通信する（または発行（publish）する）。PID2機能ブロック122dは、出力204の引用者であり、したがって、出力204を介して発行されたデータを受信する。

#### 【0086】

図14に戻ると、機能ブロック122a~eは、ループを形成するように結合または接続される。特に、AO1機能ブロック122eの出力は、PID2機能ブロック122dの入力に結合または接続されて、PID2-AO1ループ232を形成する。さらに、PID2機能ブロック122dの出力は、PID1機能ブロック122bの入力に接続されて、PID1-PID2ループ234を形成する。図示されている例では、PID1-PID2ループ234は、PID2-AO1ループ232よりも少ない頻度で実行される（例えば、より長いループ実行周期、またはより遅いループ実行レートを有する）。

#### 【0087】

ループ232のループ実行周期は、機能ブロック122c~eのブロック走査レート（block scan rate）（BSR）に基づき、そして、ループ234のループ実行周期は、機能ブロック122a~bのブロック走査レートに基づく。ブロック走査レートは、機能ブロックが、別の機能ブロックに情報を通信または発行する頻度を定義する。例えば、フィールド装置112aが、2000msごとにAI1機能ブロック122aを実行してデジタルデータバス114上に情報を発行（publish）する場合、AI1機能ブロック122aのブロック走査レートは2000msである。フィールド装置（またはコントローラ）が、それぞれの機能ブロックを実行するために必要とする時間は、本明細書ではブロック実行時間と呼ばれる。例えば、フィールド装置112aがAI1機能ブロック122aを実行するために20ミリ秒（ms）を必要とする場合、AI1機能ブロック122aのブロック実行時間は20msである。機能ブロック122a~eにそれぞれ関連付けられた、ブロック実行時間 $t_{E1}$ 、 $t_{E2}$ 、 $t_{E3}$ 、 $t_{E4}$ 、および $t_{E5}$ は、例として、図16および図17に示されている。ブロック走査レートは、一般に、機能ブロック間で異なる。図示されている例では、機能ブロック122aおよび122bに関連付けられたブロック走査レートは、機能ブロック122c~eに関連付けられたブロック走査レートよりも少ない。結果として、PID1-PID2ループ234は、PID2-AO1ループ232よりも、少ない頻度で実行される（例えば、より遅いループ実行レート、より長いループ周期など）。

10

20

30

40

50

## 【 0 0 8 8 】

図 1 6 を参照すると、例示的な実行シーケンスダイアグラム 4 0 0 は、図 1 4 のループ 2 3 2 の 2 つの連続したループ実行中における、機能ブロック 1 2 2 c ~ e の実行の間のタイミング関係を示す。A I 2 機能ブロック 1 2 2 c の実行は参照番号 4 0 2 によって示され、P I D 2 機能ブロック 1 2 2 d の実行は参照番号 4 0 4 によって示され、A O 1 機能ブロック 1 2 2 e の実行は参照番号 4 0 6 によって示されている。マクロサイクルは、本明細書では、ループの 1 回の実行（例えば、ループ 2 3 2 の 1 回の実行）を意味するために使用される。ループを実行するために必要とされる時間は、通常、最も頻度が少ないブロック走査レートを有する、ループ内の機能ブロックに基づく。公知のシステムでは、マクロサイクルに関連付けられた各機能ブロックは、マクロサイクルの間に 1 回だけ実行されなければならない。例えば、図 1 6 で、機能ブロック 1 2 2 c ~ e のそれぞれは、5 0 0 ミリ秒の周期を有するマクロサイクル 4 0 8 の間に 1 回だけ実行されるものとして示されている。公知のシステムでは、同じセグメント上のフィールド装置（例えば、デジタルデータバス 1 1 4 上のフィールド装置 1 1 2 a ~ c）は、それらのそれぞれの機能ブロックを、同じマクロサイクルに基づいて実行しなければならない。すなわち、公知のシステムの公知の設計ガイドラインは、1 つのセグメント（例えば、デジタルデータバス 1 1 4）上で異なるマクロサイクルを実行することを不可として指定している。より頻度が少ない走査レート（例えば、2 秒の走査レート）を有する別の機能ブロックがループ 2 3 2 内に導入される場合、すべての機能ブロック（例えば、機能ブロック 1 2 2 c ~ e、および 2 秒の走査レートを有する追加の機能ブロック）の 1 回の実行をマクロサイクル 4 0 8 の間に含むように、マクロサイクル 4 0 8 の周期は増加させられなければならない。

10

20

## 【 0 0 8 9 】

図 1 7 は、図 1 4 のループ 2 3 4 の 2 つの連続したループ実行中における、機能ブロック 1 2 2 a ~ b の実行の間のタイミング関係を表す、別の例示的な実行シーケンス図 5 0 0 を示す。A I 1 機能ブロック 1 2 2 a の実行は参照番号 5 0 2 によって示され、機能ブロック 1 2 2 b の実行は参照番号 5 0 4 によって示されている。機能ブロック 1 2 2 a ~ b は、機能ブロック 1 2 2 c ~ e よりも頻度が少ないブロック走査レートを有するので、ループ 2 3 4 は、ループ 2 3 2 よりも相対的に少ない頻度で実行される。図示されている例では、ループ 2 3 4 に関連付けられたマクロサイクル 5 0 2 は、ループ 2 3 4 のより少ない頻度の実行に対応するように、2 0 0 0 ミリ秒の期間 ( d u r a t i o n ) を有する。

30

## 【 0 0 9 0 】

図 1 8 は、異なるブロック走査レートを有する機能ブロック（例えば、図 1 3 ~ 図 1 4、図 1 6、および図 1 7 の機能ブロック 1 2 2 a ~ e）を同じセグメント（例えば、デジタルデータバス 1 1 4）上で実行するための、公知の方法に従って実施される、例示的な実行シーケンス図 6 0 0 を示す。具体的には、公知の方法に従って、機能ブロック 1 2 2 a ~ e を同じセグメント（例えば、デジタルデータバス 1 1 4）上で実行するために、機能ブロック 1 2 2 a ~ e の最も遅いブロック走査レートに基づいて、2 0 0 0 m s の周期を有するマクロサイクル 6 0 2 が選択される。このようにして、機能ブロック 1 2 2 a ~ e のそれぞれが、マクロサイクル 6 0 2 ごとに 1 回実行される。例示的な実行シーケンス 6 0 0 を実施するために、ループ 2 3 2 のループ実行周期がループ 2 3 4 のループ実行周期と等しくなるように、機能ブロック 1 2 2 c ~ e のブロック走査レートは減少させられる。

40

## 【 0 0 9 1 】

機能ブロック 1 2 2 c ~ e のブロック走査レートを減少させることにより、機能ブロック 1 2 2 c ~ e と機能ブロック 1 2 2 a ~ b とを同じセグメント上で実行することが可能になるが、機能ブロック 1 2 2 c ~ e のブロック走査レートを減少させることにより、機能ブロック 1 2 2 c ~ e を機能ブロック 1 2 2 a ~ b のブロック走査レートよりも速く実行することはできなくなる。いくつかの実施においては、機能ブロック 1 2 2 c ~ e を、機能ブロック 1 2 2 a ~ b のブロック実行周期よりも速く実行することが有利な場合があ

50

る。例えば、A I 2 機能ブロック 1 2 2 c が、フィールド装置 1 1 2 b ( 図 1 3 ) に圧力測定値を取得させる場合、図 1 6 に示すように 5 0 0 m s の間隔で A I 2 機能ブロック 1 2 2 c を実行することは、フィールド装置 1 1 2 b が相対的に高い分解能 ( 例えば、高い細分性 ) で複数の圧力測定値を取得して、デジタルデータバス 1 1 4 上に発行 ( p u b l i s h ) することを可能にする。図 1 6 に示すように、圧力測定値をより頻繁に ( 例えば、相対的に高い分解能で ) 取得することは、フィールド装置 1 1 2 b が、例えば、5 0 0 m s の範囲内で発生するブリップ、スパイク、またはその他の比較的高い頻度の圧力挙動を捕捉することを可能にする。対照的に、公知の方法を使用して例示的な実行シーケンス 6 0 0 を生成するために A I 2 機能ブロック 1 2 2 c のブロック走査レートを遅くすると、フィールド装置 1 1 2 b は、2 0 0 0 m s 未満の範囲内で発生するブリップ、スパイク、またはその他の比較的高い頻度の圧力挙動を捕捉することができなくなる。

10

## 【 0 0 9 2 】

すべてのループのループ実行周期を同じにすることによって 1 つのセグメント上で複数のループを実行するための、例示的な実行シーケンス 6 0 0 を生成するために使用される、公知の方法とは異なる本明細書に記載された例示的な方法および装置は、異なるループ実行周期を有する複数のループを同じセグメント上でスケジュールすることを可能にする。図 1 9 を参照すると、本明細書に記載された例示的な方法および装置に従って実施される例示的な実行シーケンス図 7 0 0 は、図 1 4 ~ 図 1 6 のループ 2 3 2 および 2 3 4 が、それらのそれぞれのループ実行周期に従って実行されることを可能にする。このようにして、より速いブロック走査レートを有する機能ブロックは、より遅いブロック走査レートを有する、同じセグメント上の機能ブロックよりも、より速いレートで実行されることが可能となる。

20

## 【 0 0 9 3 】

デジタルデータバス 1 1 4 上で、ループ 2 3 2 および 2 3 4 を、それらのそれぞれのレートで実行することを可能にするために、本明細書に記載された例示的な方法および装置は、機能ブロック 1 2 2 a ~ e の複数の機能ブロック実行 4 0 2、4 0 4、4 0 6、5 0 2、および 5 0 4 として図 1 9 に示すスケジュール 7 0 2 ( すなわち、機能ブロック実行スケジュール ) を生成する。図示されている例では、スケジュール 7 0 0 は、時間  $t_0$  において始まるスケジュール開始時間 7 0 4 を有する。機能ブロック実行 4 0 2、4 0 4、4 0 6、5 0 2、および 5 0 4 のそれぞれは、スケジュール開始時間 7 0 4 に相対的なそれぞれの開始時間オフセット ( 例えば、オフセット ) において始まる。A I 2 機能ブロック 1 2 2 c の機能ブロック実行 4 0 2 のうちの 1 つが始まる、スケジュール開始時間 7 0 4 に相対的な時を示すために、例示的な開始時間オフセット 7 0 6 が図 1 9 に示されている。

30

## 【 0 0 9 4 】

スケジュール 7 0 0 は、機能ブロック 1 2 2 a ~ e のブロック走査レートと、モジュール 1 2 0 のモジュール実行周期 (  $T_{ME}$  ) とに基づいて決定される。モジュール実行周期 (  $T_{ME}$  ) は、モジュール 1 2 0 内の機能ブロック ( 例えば、機能ブロック 1 2 2 a ~ e のうちの 1 つ ) に関連付けられた、最も遅い、または最も少ない頻度のブロック走査レートの逆数に等しい。例えば、モジュール 1 2 0 のモジュール走査周期 (  $T_{ME}$  ) は 2 0 0 0 m s であり、その理由は、A I 1 機能ブロック 1 2 2 a と P I D 1 機能ブロック 1 2 2 b は 1 走査 / 2 0 0 0 m s のブロック走査レートを有し、これは、モジュール 1 2 0 内の機能ブロック 1 2 2 c ~ e の 1 走査 / 5 0 0 m s のブロック走査レートよりも遅いからである。

40

## 【 0 0 9 5 】

本明細書に記載された例示的な方法および装置は、各機能ブロック ( 例えば、機能ブロック 1 2 2 a ~ e のそれぞれ ) について、1 つ以上の開始時間オフセットを生成するように構成される。機能ブロックについての開始時間オフセットの数は、その機能ブロックのブロック未加工実行周期 ( b l o c k r a w e x e c u t i o n p e r i o d ) (  $T_{BRE}$  ) と、その機能ブロックを実行するように構成されたフィールド装置に関連付けられた、最も遅いブロック未加工実行周期 ( s l o w e s t b l o c k r a w e x e

50

cution period) ( $T_{SBE}$ )とに基づいて決定される。ブロック未加工実行周期 ( $T_{BRE}$ ) は、スケジュール (例えば、図 19 のスケジュール 702) に関係なく、機能ブロックの実行の間の時間の長さを定義し、次の式 1 に従って定義され得る。

$$(式1) \quad T_{BRE} = T_{ME} \times F_{BSR}$$

上記式 1 に示すように、特定の機能ブロックのブロック未加工実行周期 ( $T_{BRE}$ ) は、その機能ブロックを含むモジュールのモジュール実行周期 ( $T_{ME}$ ) に、その機能ブロックのブロック走査レート因子 (block scan rate factor) ( $F_{BSR}$ ) を掛けることによって決定される。ブロック走査レート因子 ( $F_{BSR}$ ) は、機能ブロックの 2 つの連続した実行の開始時間の間のモジュール実行周期 ( $T_{ME}$ ) の数量に等しく、次の式 2 を使用して決定され得る。

$$(式2) \quad F_{BSR} = T_{ME} / BSR$$

上記の式 2 と、図 19 とを参照すると、モジュール 120 (図 13) のモジュール実行周期 ( $T_{ME}$ ) が 2000ms であり、かつ、機能ブロック 122a が 2000ms ごとに実行されるように構成されている場合 ( $BSR = 2000ms$ )、AI1 機能ブロック 122a のブロック走査レート因子 ( $F_{BSR}$ ) は 1 に等しく、なぜなら、AI1 機能ブロック 122a の 2 つの連続した実行の開始の間に、1 つのモジュール実行周期 ( $T_{ME}$ ) が経過するからである。図 19 で、AI2 機能ブロック 122c は、500ms ごとに実行されるとして示されている ( $BSR = 500ms$ )。したがって、AI2 機能ブロック 122c のブロック走査レート因子 ( $F_{BSR}$ ) は 4 分の 1 (0.25) に等しく、なぜなら、AI2 機能ブロック 122c の 2 つの連続した実行の開始の間に、モジュール実行周期 ( $T_{ME}$ ) の 4 分の 1 のみが経過するからである。

【0096】

上記の式 1 を再び参照すると、モジュール 120 のモジュール実行周期 ( $T_{ME}$ ) が 2000ms であり、かつ、AI1 機能ブロック 122a のブロック走査レート因子 ( $F_{BSR}$ ) が 1 に等しい場合、AI1 機能ブロック 122a のブロック未加工実行周期 ( $T_{BRE}$ ) は 2000ms に等しく、これは、AI1 機能ブロック 122a が 2000ms ごとに実行されることを示している。しかし、AI2 機能ブロック 122c のブロック未加工実行周期 ( $T_{BRE}$ ) は 500ms に等しく、なぜなら、AI2 機能ブロック 122c のブロック走査レート因子 ( $F_{BSR}$ ) は 4 分の 1 だからである (例えば、 $500ms (T_{BRE}) = 2000ms (T_{ME}) \times 1/4 (F_{BSR})$ )。

【0097】

上記の式 1 に基づいてブロック未加工実行周期 ( $T_{BRE}$ ) が決定された後で、ブロック未加工実行周期 ( $T_{BRE}$ ) は、図 20 の丸めテーブル 800 に基づいて切り上げられる。ブロック未加工実行周期 ( $T_{BRE}$ ) が切り上げられるのは、スケジュール 702 (図 19) の複数の繰り返しの後で、機能ブロック実行 (例えば、図 19 の機能ブロック実行 402、404、406、502、および 504) が時間に関してスキューしないことを確実にするため、ならびに、各機能ブロックについて必要とされる開始時間オフセットの数量 (quantity of start time offsets) ( $Q_s$ ) を、丸められたブロック未加工実行周期 (rounded block raw execution period) ( $RT_{BRE}$ ) の境界に基づいて決定するためである。丸めテーブル 800 に示すように、機能ブロックのブロック未加工実行周期 ( $T_{BRE}$ ) が 0ms と 500ms との間である場合、ブロック未加工実行周期 ( $T_{BRE}$ ) は、丸められたブロック未加工実行周期 ( $RT_{BRE}$ ) に切り上げられる。同様に、機能ブロックのブロック未加工実行周期 ( $T_{BRE}$ ) が、500ms と 1000ms との間、または 1000ms と 2000ms との間、または 2000ms よりも大きい場合、ブロック未加工実行周期 ( $T_{BRE}$ ) は、それぞれ、1000ms、2000ms、または 4000ms の丸められたブロック未加工実行周期 ( $RT_{BRE}$ ) 値に切り上げられる。

【0098】

図 19 に戻ると、スケジュール 702 は、サブスケジュール 710 および 712 を使用して実施される。図示されている例では、サブスケジュール 710 は 500ms の周期を

10

20

30

40

50

有し、そして、図 1 4 および図 1 6 のループ 2 3 2 のループ実行に関連付けられている。サブスケジュール 7 1 2 は 2 0 0 0 m s の周期を有し、そして、図 1 4 および図 1 7 のループ 2 3 4 のループ実行に関連付けられている。スケジュール 7 0 2 の生成の間は、サブスケジュール 7 1 0 が最初に生成され、そして次に、サブスケジュール 7 1 0 は、2 0 0 0 m s のスケジュール 7 0 2 を埋めるために、図 1 9 に示すように 3 回複製される。サブスケジュール 7 1 0 を生成した後は、周期が次に最も短いサブスケジュール（例えば、サブスケジュール 7 1 2）が生成される。サブスケジュール 7 1 0 および 7 1 2 を生成した後で、サブスケジュール 7 1 0 および 7 1 2 は、スケジュール 7 0 2 を生成するためにマージされる。

#### 【 0 0 9 9 】

図 1 9 に示すように、サブスケジュール 7 1 0 の間に、実行 4 0 2、4 0 4、および 4 0 6 のそれぞれは 1 回発生する。したがって、機能ブロック 1 2 2 c ~ e のそれぞれは、サブスケジュール 7 1 0 について開始時間オフセットを 1 つだけ必要とする。同様に、サブスケジュール 7 1 2 の間に、機能ブロック 1 2 2 a ~ b のそれぞれは、1 回のみ実行される。したがって、機能ブロック 1 2 2 a ~ b のそれぞれは、サブスケジュール 7 1 2 について開始時間オフセットを 1 つだけ必要とする。

#### 【 0 1 0 0 】

機能ブロック（例えば、図 1 3 および図 1 4 の機能ブロック 1 2 2 a ~ e のうちの 1 つ）によって必要とされる、サブスケジュール（例えば、サブスケジュール 7 1 0 またはサブスケジュール 7 1 2）についての、開始時間オフセット（例えば、開始時間オフセット 7 0 6）の数量（ $Q_S$ ）は、その機能ブロックの、丸められたブロック未加工実行周期（ $RT_{BRE}$ ）と、その機能ブロックを実行するフィールド装置（例えば、図 1 3 のフィールド装置 1 1 2 a ~ c のうちの 1 つ）に関連付けられた、最も遅い丸められたブロック未加工実行周期（*slowest rounded block raw execution period*）（ $RT_{SRE}$ ）とに基づいて決定される。機能ブロックについての、開始時間オフセットの数量（ $Q_S$ ）は、次の式 3 を使用して決定されてもよい。

$$(式 3) \quad Q_S = RT_{SRE} / RT_{BRE}$$

上記の式 3 に示すように、機能ブロックについての、開始時間オフセットの数量（ $Q_S$ ）は、その機能ブロックを実行するフィールド装置に関連付けられた、最も遅い丸められたブロック未加工実行周期（ $RT_{SRE}$ ）を、その機能ブロックの、丸められたブロック未加工実行周期（ $RT_{BRE}$ ）で割ることによって決定される。

#### 【 0 1 0 1 】

P I D 2 機能ブロック 1 2 2 d についての、開始時間オフセットの数量（ $Q_S$ ）を決定するために式 3 を使用することは、最初に、P I D 2 機能ブロック 1 2 2 d を実行するフィールド装置 1 1 2 c に関連付けられた、最も遅い丸められたブロック未加工実行周期（ $RT_{SRE}$ ）を決定することを含む。図示されている例では、フィールド装置 1 1 2 c によって実行されるのは、P I D 2 および A O 1 機能ブロック 1 2 2 d ~ e のみである。したがって、フィールド装置 1 1 2 c に関連付けられた最も遅い丸められたブロック未加工実行周期（ $RT_{SRE}$ ）は、より遅い、丸められたブロック未加工実行周期（ $RT_{BRE}$ ）を有する機能ブロック 1 2 2 d ~ e のうちの 1 つの、丸められたブロック未加工実行周期（ $RT_{BRE}$ ）に等しい。機能ブロック 1 2 2 d ~ e の丸められたブロック未加工実行周期（ $RT_{BRE}$ ）は、両方とも 5 0 0 m s に等しいので、最も遅い丸められたブロック未加工実行周期（ $RT_{SRE}$ ）は 5 0 0 m s に等しく設定される。

#### 【 0 1 0 2 】

フィールド装置 1 1 2 c に関連付けられた、最も遅い丸められたブロック未加工実行周期（ $RT_{SRE}$ ）を決定した後は、式 3 を使用して、5 0 0 m s（フィールド装置 1 1 2 c に関連付けられた、最も遅い丸められたブロック未加工実行周期（ $RT_{SRE}$ ）を、5 0 0 m s（P I D 2 機能ブロック 1 2 2 d の、丸められたブロック未加工実行周期（ $RT_{BRE}$ ））で割ることによって、P I D 2 機能ブロック 1 2 2 d のために必要とされる開始時間オフセットの数量（ $Q_S$ ）が決定されてもよい。この割り算操作は、サブスケジュール

10

20

30

40

50

710内で実行されるPID2機能ブロック122dに必要な開始時間オフセットの数量( $Q_S$ )が1であることを示す。

【0103】

図21の、例示的な開始時間オフセットの数量テーブル900は、図19のサブスケジュール712の間に、実行のために機能ブロック122a~bによって必要とされる開始時間オフセットの数量( $Q_S$ )と、図19のサブスケジュール710の間に、実行のために機能ブロック122c~eによって必要とされる開始時間オフセットの数量( $Q_S$ )とを示す。開始時間オフセットの数量テーブル900は、ブロック走査レート因子( $F_{BSR}$ )、ブロック未加工実行周期( $T_{BRE}$ )、および、丸められたブロック未加工実行周期( $RT_{BRE}$ )も示す。

10

【0104】

図22は、モジュール1020の機能ブロック1022a~eに基づいてプロセス制御システムを実施するために複数のフィールド装置1012a~cがデジタルデータバス1014に通信可能に結合された、別の例示的な機能ブロック構成1000を示す。図21の開始時間オフセットの数量テーブル900に示すように、開始時間オフセットを1つだけ必要とする図13および図14の機能ブロック122a~eとは異なり、図22に示す機能ブロック1022a~eのうちのいくつかは、以下で述べるように、複数の開始時間オフセットを必要とする。複数のフィールド装置1012a~cは、図13のフィールド装置112a~cに類似しているか、またはそれらと同一であり、そして、デジタルデータバス1014は、図13のデジタルデータバス114に類似しているか、またはそれと同一である。図22に示すように、フィールド装置1012aは、AI11機能ブロック1022aと、AI12機能ブロック1022cと、AO11機能ブロック1022eとによって定義される機能を実行し、フィールド装置1012bは、PID12機能ブロック1022bによって定義される機能を実行し、フィールド装置1012cは、PID13機能ブロック1022dによって定義される機能を実行する。

20

【0105】

AI11機能ブロック1022aは、図13および図14のAI1機能ブロック122aに類似しているか、またはそれと同一であり、PID12機能ブロック1022bは、図13および図14のPID1機能ブロック122bに類似しているか、またはそれと同一であり、AI12機能ブロック1022cは、図13および図14のAI2機能ブロック122cに類似しているか、またはそれと同一であり、PID13機能ブロック1022dは、図13および図14のPID2機能ブロック122dに類似しているか、またはそれと同一であり、AO11機能ブロック1022eは、図13および図14のAO1機能ブロック122eに類似しているか、またはそれと同一である。例えば、AI11機能ブロック1022aおよびPID12機能ブロック1022bのブロック走査レートは、2000msに等しい。同様に、機能ブロック1022c~eのブロック走査レートは、500msに等しい。図示されていないが、機能ブロック1022a~eの間の接続は、図14に示す機能ブロック122a~eの間の接続と同一である。

30

【0106】

図23の、別の例示的な、開始時間オフセットの数量テーブル1100は、機能ブロック1022a~e(図22)のそれぞれについての、それらのそれぞれのブロック走査レート因子( $F_{BSR}$ )、ブロック未加工実行周期( $T_{BRE}$ )、および、丸められたブロック未加工実行周期( $RT_{BRE}$ )に基づいて決定される開始時間オフセットの数量( $Q_S$ )を示す。機能ブロック1022a~eのブロック未加工実行周期( $T_{BRE}$ )は、上記の式1に基づいて決定され、式1は、さらに、ブロック走査レート因子( $F_{BSR}$ )と、モジュールのモジュール実行周期( $T_{ME}$ )とに基づく。ブロック未加工実行周期( $T_{BRE}$ )のそれぞれは、次に、機能ブロック1022a~eのそれぞれの、丸められたブロック未加工実行周期( $RT_{BRE}$ )を決定するために丸められる。

40

【0107】

機能ブロック1022a~eのそれぞれについての、開始時間オフセットの数量( $Q_S$ )

50

)が、次に、上記の式3に基づいて決定される。図23に示すように、参照番号1102は、AI12機能ブロック1022c (PDT11/FFAI12) についての、開始時間オフセットの数量( $Q_S$ )が4に等しいことを示す。AI12機能ブロック1022c についての、開始時間オフセットの数量( $Q_S$ )を決定するには、フィールド装置1012aに関連付けられた、最も遅い丸められたブロック未加工実行周期( $RT_{SRE}$ )が最初に決定される。図22に示されている例では、フィールド装置1012aは、AI11機能ブロック1022aと、AI12機能ブロック1022cと、AO11機能ブロック1022eとを実行するように構成されている。したがって、最も遅い丸められたブロック未加工実行周期( $RT_{SRE}$ )は、AI11機能ブロック1022aの、丸められたブロック未加工実行周期( $RT_{BRE}$ )と、AI12機能ブロック1022cの、丸められたブロック未加工実行周期( $RT_{BRE}$ )と、AO11機能ブロック1022eの、丸められたブロック未加工実行周期( $RT_{BRE}$ )とのうちで最も遅いものである。図23に示すように、AI11機能ブロック1022a (PDT11/FFAI11) の丸められたブロック未加工実行周期( $RT_{BRE}$ )は2000msであり、AI12機能ブロック1022c (PDT11/FFAI12) の丸められたブロック未加工実行周期( $RT_{BRE}$ )は500msであり、AO11機能ブロック1022e (PDT11/FFAO11) の丸められたブロック未加工実行周期( $RT_{BRE}$ )は500msである。したがって、フィールド装置1012aに関連付けられた最も遅い丸められたブロック未加工実行周期( $RT_{SRE}$ )は2000msである。上記の式3に従って、AI12機能ブロック1022cによって必要とされる開始時間オフセットの数量( $Q_S$ )を決定するために、2000ms (フィールド装置1012aに関連付けられた、最も遅い丸められたブロック未加工実行周期( $RT_{SRE}$ ))が500ms (AI12機能ブロック1022cの、丸められたブロック未加工実行周期( $RT_{BRE}$ ))で割られて、参照番号1102によって示されている値4が計算される。

#### 【0108】

テーブル1100は、さらに、AI11機能ブロック1022a (PDT11/FFAI11) によって必要とされる開始時間オフセットの数量( $Q_S$ )が、参照番号1104によって示されるように1に等しいということも示している。図示されている例では、AI11機能ブロック1022aは1つの開始時間オフセットを有し、AI12機能ブロック1022cは4つの開始時間オフセットを有し、その理由は、AI11機能ブロック1022aの丸められたブロック未加工実行周期( $RT_{BRE}$ ) (2000ms)は、フィールド装置1012aに関連付けられた最も遅い丸められたブロック未加工実行周期( $RT_{SRE}$ ) (2000ms)に等しく、一方、AI12機能ブロック1022cの丸められたブロック未加工実行周期( $RT_{BRE}$ ) (500ms)は、フィールド装置1012aに関連付けられた、最も遅い丸められたブロック未加工実行周期( $RT_{SRE}$ ) (2000ms)の4分の1であるからである。したがって、フィールド装置1012aは、AI11機能ブロック1022aの各実行につき、AI12機能ブロック1022cを4回実行する必要がある。

#### 【0109】

図24は、スケジュール1202 (すなわち、機能ブロック実行スケジュール)と、それぞれの機能ブロック1022a~e (図22) の実行1204、1206、1208、1210、および1212とを示す、別の例示的な実行シーケンス図1200である。スケジュール1202は、PID13機能ブロック1022dに関連付けられた500msのサブスケジュール1214と、機能ブロック1022a~cおよび1022eに関連付けられた2000msのサブスケジュール1216とを使用して生成される。図示されているように、500msのサブスケジュール1214は、スケジュール1202の間に4回繰り返され、2000msのサブスケジュール1216は、スケジュール1202の間に1回発生する。

#### 【0110】

図23のテーブル1100に従って、2000msのサブスケジュール1216内で、

10

20

30

40

50

AI11機能ブロック1022aは1つの開始時間オフセット(開始時間オフセット $t_0$ )を有し、AI12機能ブロック1022cは4つの開始時間オフセット(開始時間オフセット $t_1$ 、 $t_5$ 、 $t_9$ 、および $t_{13}$ )を有し、AO11機能ブロック1022eは4つの開始時間オフセット(開始時間オフセット $t_3$ 、 $t_7$ 、 $t_{11}$ 、および $t_{15}$ )を有し、PID12機能ブロック1022bは1つの開始時間オフセット(開始時間オフセット $t_1$ )を有する。同様に、図23のテーブル1100に従って、500msのサブスケジュール1214内でPID13機能ブロック1022dは1つの開始時間オフセット(開始時間オフセット $t_2$ )を有する。

【0111】

機能ブロックの開始時間オフセットは、次の式4に従って決定されてもよい。

$$(式4) \quad t_s = N_{Seq} \times T_{BRE} + t_{DA}$$

上記式4で示したように、機能ブロックについての開始時間オフセット( $t_s$ )は、計算されるべき現在の開始時間オフセットの順序番号( $N_{Seq}$ )に機能ブロックのブロック未加工実行周期( $T_{BRE}$ )を掛け、結果として得られた積を機能ブロックのデータ利用可能時間( $t_{DA}$ )に足すことによって決定される。順序番号( $N_{Seq}$ )は、サブスケジュール(例えば、図19のサブスケジュール710または712のうちの1つ)の間の、機能ブロックの特定の実行インスタンスを指す。例えば、図24で、AI12機能ブロック1022cは、2000msのサブスケジュール1216の間に4つの実行インスタンスを有する(すなわち、開始時間オフセットの数量( $Q_s$ )は4に等しく、順序番号( $N_{Seq}$ )は0~3)。機能ブロックのデータ利用可能時間( $t_{DA}$ )は、機能ブロックを実行するために必要とされるデータが、別の機能ブロックによって利用可能にされる機能ブロックのブロック未加工実行周期( $T_{BRE}$ )の間の時間を意味する。例えば、第1の機能ブロックのデータ利用可能時間( $t_{DA}$ )は、第2の機能ブロックからのデータが、第1の機能ブロックを実行するフィールド装置にとって利用可能になる時(例えば、そのデータがデジタルデータバス1014上に発行(publish)される時)に基づいて決定される。フィールド装置が第1の機能ブロックを実行するために、第1の機能ブロックが、第2の機能ブロックからのデータも、その他のいかなる機能ブロックからのデータも必要としない場合、第1の機能ブロックのデータ利用可能時間( $t_{DA}$ )は0に設定される。さらなる例として、機能ブロックのブロック未加工実行周期( $T_{BRE}$ )が500msであり、500msのブロック未加工実行周期( $T_{BRE}$ )の開始を基準として125msにおいて、別の機能ブロックによってデータがその機能ブロックにとって利用可能にされる場合、その機能ブロックのデータ利用可能時間( $t_{DA}$ )は125msである。

【0112】

式4と、AI12機能ブロック1022cの実行1206とを参照すると、オフセット開始時間( $t_s$ ) $t_1$ 、 $t_5$ 、 $t_9$ 、および $t_{13}$ は、以下のように決定されてもよい。図24に示すように、AI12機能ブロック1022cのブロック未加工実行周期( $T_{BRE}$ )が500msである場合、2000msのサブスケジュール1216の間の開始時間オフセットの数量( $Q_s$ )は4(すなわち、順序番号( $N_{Seq}$ )は0~3)であり、なぜなら、2000msのサブスケジュール1216のサブスケジュール実行周期をAI12機能ブロック1022cの500msのブロック未加工実行周期( $T_{BRE}$ )で割ると4に等しいからである。さらに、AI12機能ブロック1022cのデータ利用可能時間( $t_{DA}$ )は125msである。したがって、上記の式4を使用すると、第1の順序番号( $N_{Seq}$ )に対応する開始時間オフセット( $t_s$ ) $t_1$ は125msに等しく、第2の順序番号( $N_{Seq}$ )に対応する開始時間オフセット( $t_s$ ) $t_5$ は625msに等しく、第3の順序番号( $N_{Seq}$ )に対応する開始時間オフセット( $t_s$ ) $t_9$ は1125msに等しく、第4の順序番号( $N_{Seq}$ )に対応する開始時間オフセット( $t_s$ ) $t_{13}$ は1625msに等しい。

【0113】

図25は、図13のフィールド装置112a~cと、図13の機能ブロック122a~eと、機能ブロック122a~eに対応する開始時間オフセットとを表すオブジェクトの関係を示す、例示的なオブジェクト指向ソフトウェア実行環境1300である。図1

10

20

30

40

50

3のフィールド装置112aに対応するPDT1装置オブジェクト1302aは、FFAI1機能オブジェクト1304aと、FFPID1機能オブジェクト1304bとに結合される。FFAI1機能オブジェクト1304aは図13のAI1機能ブロック122aに対応し、FFPID1機能オブジェクト1304bは図13のPID1機能ブロック122bに対応する。図13のフィールド装置112bに対応するPDT2装置オブジェクト1302bは、AI2機能ブロック122b(図13)に対応するFFAI2機能オブジェクト1304cと、PID2機能ブロック122c(図13)に対応するFFPID2機能オブジェクト1304dとに結合される。図13のフィールド装置112cに対応するPDT3装置オブジェクト1302cは、AO1機能ブロック122e(図13)に対応するFFAO1機能オブジェクト1304eに結合される。

10

## 【0114】

図21の、開始時間オフセットの数量テーブル900に従って、機能オブジェクト1304a~eのそれぞれは、図25に示す複数の開始時間オフセット1306a~eのうちの該当する1つを有する。開始時間オフセットのそれぞれは、データ構造を使用して、メモリ(例えば、図13のワークステーション102のメモリ)内に記憶されてもよい。図25に示されている例では、開始時間オフセット1306に対応する例示的な開始時間オフセットデータ構造1310が、開始時間オフセット1306を、FBS{順序番号,インデックス値,開始時間オフセット}という形式で記憶する。この例では、順序番号は、サブスケジュール(例えば、図19のサブスケジュール710または712のうちの1つ)の間の機能ブロックの特定の実行インスタンスを指し、インデックス値は、開始時間オフセットをそれぞれの機能ブロックと関連付けるために使用されてもよく、そして、開始時間オフセットはそれぞれの機能ブロックの開始時間オフセット(例えば、開始時間オフセット1306b)を意味する。

20

## 【0115】

図26は、図22のフィールド装置1012a~cと、図22の機能ブロック1022a~eと、機能ブロック1022a~eに対応する開始時間オフセットとを表すオブジェクトの間の関係を示す、別の例示的なオブジェクト指向ソフトウェア実行環境1400である。図22のフィールド装置1012aに対応するPDT11装置オブジェクト1402aは、AI11機能ブロック1022a(図22)に対応するFFAI11機能オブジェクト1404aと、AI12機能ブロック1022c(図22)に対応するFFAI12機能オブジェクト1404bと、図22のAO11機能ブロック1022eに対応するFFAO11機能オブジェクト1404cとに結合される。図22のフィールド装置1012bに対応するPDT12装置オブジェクト1402bは、PID12機能ブロック1022b(図22)に対応するFFPID12機能オブジェクト1404dに結合される。図22のフィールド装置1012cに対応するPDT13装置オブジェクト1402cは、PID13機能ブロック1022d(図22)に対応するFFPID13機能オブジェクト1404eに結合される。

30

## 【0116】

図23の、開始時間オフセットの数量テーブル1100に従って、機能オブジェクト1404a~eのそれぞれは、図26に示す複数の開始時間オフセット1406a~kのうちの該当する1つ以上を有する。FFAO11機能オブジェクト1404cに対応する例示的な開始時間オフセットデータ構造1410は、開始時間オフセット1406f~iを有する開始リスト1412を記憶する。オフセットデータ構造1410と、図24の例示的な実行シーケンス図1200とを参照すると、開始時間オフセット1406fは、順序番号1と $t_3$ の開始時間オフセットとに関連付けられ、開始時間オフセット1406gは、順序番号2と $t_7$ の開始時間オフセットとに関連付けられ、開始時間オフセット1406hは順序番号3と $t_{11}$ の開始時間オフセットとに関連付けられ、開始時間オフセット1406iは順序番号4と $t_{15}$ の開始時間オフセットとに関連付けられる。

40

## 【0117】

図27は、AO11機能ブロック1022e(図22)に関連付けられた開始リスト1

50

4 1 2 ( 図 2 6 ) が、ワークステーション 1 0 2 ( 図 1 3 ) からフィールド装置 1 0 1 2 a ( 図 2 2 ) にコピーされる例示的な方法を示す。フィールド装置 1 0 1 2 a ~ c がそれらのそれぞれの機能ブロック 1 0 2 2 a ~ e ( 図 2 2 ) に対応する機能を、図 2 4 のスケジュール 1 2 0 2 の間のスケジュールされた時間において実行することを可能にするために、ワークステーション 1 0 2 は、フィールド装置 1 0 1 2 a ~ c に開始時間オフセット ( 例えば、開始時間オフセット 1 4 0 6 f ~ i ) を有する開始リスト ( 例えば、開始リスト 1 4 1 2 ) をダウンロードする。図 2 7 に示すように、ワークステーション 1 0 2 は、A O 1 1 機能ブロック 1 0 2 2 e を含む図 2 2 のモジュール 1 0 2 0 を記憶する。図示されていないが、ワークステーション 1 0 2 は図 2 2 のその他の機能ブロック 1 0 2 2 a ~ d も記憶する。図示されている例では、ワークステーション 1 0 2 は、図 2 6 の開始時間  
10  
オフセット 1 4 0 6 f ~ i を有する開始リスト 1 4 1 2 を記憶する。図示されていないが、ワークステーション 1 0 2 は、機能ブロック 1 0 2 2 a ~ d に対応する開始リストも記憶する。

#### 【 0 1 1 8 】

図 2 7 には、フィールド装置 1 0 1 2 a をデジタルデータバス 1 0 1 4 に通信可能に結合するように構成されたポート 1 5 0 4 を有する図 2 2 のフィールド装置 1 0 1 2 a も示されている。ポート 1 5 0 4 は、デジタルデータバス 1 0 1 4 に関連付けられたスケジュール 1 2 0 2 ( 図 2 4 ) を備える。特に、スケジュール 1 2 0 2 は、ポート 1 5 0 4 に、スケジュールの期間 ( 例えば、2 0 0 0 m s ) と、デジタルデータバス 1 0 1 4 に通信可能に結合されたすべてのその他のフィールド装置 ( 例えば、フィールド装置 1 0 1 2 a ~  
20  
c ) がそれらのそれぞれの機能 ( 例えば、図 2 2 の機能ブロック 1 0 2 2 a ~ d に対応する機能 ) を実行するように構成された時間とを指示する。

#### 【 0 1 1 9 】

図示されている例では、フィールド装置 1 0 1 2 a は、A O 1 1 機能ブロック 1 0 2 2 e に対応する A O 機能であって A O フィールド機能 ( F F ) ブロック 1 5 0 6 として示されている A O 機能を記憶する。もちろん、フィールド装置 1 0 1 2 a は、例えば、図 2 2 に示す A I 1 1 機能ブロック 1 0 2 2 a に対応する A I フィールド機能ブロックを含むより多くの機能を記憶してもよい。A O 1 1 機能ブロック 1 0 2 2 e と、A O フィールド機能ブロック 1 5 0 6 と、開始リスト 1 4 1 4 とは、A O 1 1 機能ブロック 1 0 2 2 e と A O フィールド機能ブロック 1 5 0 6 とを開始リスト 1 4 1 4 内の開始時間オフセット 1 4  
30  
0 6 f ~ i に ( 例えば、キーによって ) 関連付けるインデックス値 1 5 0 8 を記憶する。図 1 3 のワークステーション 1 0 2 またはコントローラ 1 0 6 内のスケジュール生成プロセスが開始時間オフセット 1 4 0 6 f ~ i を決定し、開始リスト 1 4 1 4 内に開始時間オフセット 1 4 0 6 f ~ i を記憶した後は、ワークステーション 1 0 2 またはコントローラ 1 0 6 は、開始時間オフセット 1 4 0 6 f ~ i を有する開始リスト 1 4 1 4 をデジタルデータバス 1 0 1 4 を介してフィールド装置 1 0 1 2 a に通信する。図 2 7 に示すように、フィールド装置 1 0 1 2 a は、次に、開始リスト 1 4 1 4 を記憶する。フィールド装置 1 0 1 2 a は、次に、開始時間オフセット 1 4 0 6 f ~ i を使用して、スケジュール 1 2 0 2 に従った適切な時に A O フィールド機能ブロック 1 5 0 6 を実行するタイミングを決定する。図示されていないが、フィールド装置 1 0 1 2 a は、A I 1 1 および A I 1 2 機能  
40  
ブロック 1 0 2 2 a および 1 0 2 2 c ( 図 2 2 ) に対応する開始リストも記憶する。

#### 【 0 1 2 0 】

図 2 8 は、本明細書に記載された例示的な方法および装置を使用して、スケジュール ( 例えば、図 1 9 および図 2 4 のスケジュール 7 0 2 および 1 2 0 2 ) を生成するように構成された複数のクラス 1 6 0 2 a ~ i を有する、例示的なオブジェクト指向プログラミング環境 1 6 0 0 を示す。D b s I n t e r f a c e D e v i c e クラス 1 6 0 2 a は、フィールド装置 ( 例えば、図 1 3 のフィールド装置 1 1 2 a ~ c、または図 2 2 のフィールド装置 1 0 1 2 a ~ c ) あるいはフィールド装置オブジェクト ( 例えば、図 2 5 のフィールド装置オブジェクト 1 3 0 2 a ~ c、または図 2 6 のフィールド装置オブジェクト 1 4  
50  
0 2 a ~ c ) と情報を交換するために提供される。例えば、D b s I n t e r f a c e D

ev i c e クラス 1 6 0 2 a は、フィールド装置またはフィールド装置オブジェクトと通信するように構成された、複数の関数またはメソッドを備えてもよい。D b s I n t e r f a c e D e v i c e クラス 1 6 0 2 a の例示的な関数は、フィールド装置 1 1 2 a ~ c または 1 0 1 2 a ~ c のそれぞれに関連付けられたブロック走査レートまたはブロック未加工実行周期 ( T <sub>BRE</sub> ) を取得する。

#### 【 0 1 2 1 】

D b s I n t e r f a c e M o d u l e クラス 1 6 0 2 b は、1 つ以上のフィールド装置 ( 例えば、図 1 3 のフィールド装置 1 1 2 a ~ c 、または図 2 2 のフィールド装置 1 0 1 2 a ~ c ) であって、それらについてのスケジュール ( 例えば、図 1 9 のスケジュール 7 0 2 、または図 2 4 のスケジュール 1 2 0 2 ) が生成されるべき 1 つ以上のフィールド装置を含む、モジュールまたはモジュールオブジェクト ( 例えば、図 1 3 のモジュール 1 2 0 、または図 2 2 のモジュール 1 0 2 0 ) と情報を交換するために提供される。D b s I n t e r f a c e M o d u l e クラス 1 6 0 2 b の例示的な関数またはメソッドは、G U I 制御システム設計インタフェースを使用して指定される機能ブロックの相互接続 ( 例えば、図 1 4 に示す機能ブロック 1 2 2 a ~ e の相互接続 ) によって定義されるモジュール ( 例えば、図 1 3 のモジュール 1 2 0 ) から、機能ブロック ( 例えば、図 1 3 および図 1 4 の機能ブロック 1 2 2 a ~ e ) の実行順序を取得するように構成されてもよい。

#### 【 0 1 2 2 】

D b s S c h e d u l e B a s e T e m p l a t e クラス 1 6 0 2 c は、スケジュールの基準テンプレート ( 例えば、図 1 9 のスケジュール 7 0 2 を生成するための基準テンプレート、または図 2 4 のスケジュール 1 2 0 2 を生成するための基準テンプレート ) を取得および / または生成するために提供される。スケジュールの基準テンプレートは、スケジュールを生成するために必要とされる、基本的な又は基礎的な枠組みを提供する。例えば、スケジュールの基準テンプレートは、機能ブロックをスケジュールすることに関連付けられた規則を含んでもよく、かつ / または、基準テンプレートは、デフォルトパラメータ ( 例えば、スケジュール期間、スケジュール期間ごとの実行など ) を指定してもよい。いくつかの例示的实施では、スケジュールのための基準テンプレートは、例えば、ワークステーション 1 0 2 内のスケジュール基準テンプレートデータベース ( 図示せず ) から取得されてもよい。

#### 【 0 1 2 3 】

D b s S c h e d u l e クラス 1 6 0 2 d は、本明細書に記載されたスケジュール ( 例えば、図 1 9 のスケジュール 7 0 2 、図 2 4 のスケジュール 1 2 0 2 、または任意のその他のスケジュール ) を生成するために提供される。D b s S u b S c h e d u l e クラス 1 6 0 2 e は、本明細書に記載されたサブスケジュール ( 例えば、図 1 9 のサブスケジュール 7 1 0 および 7 1 2 、図 2 4 のサブスケジュール 1 2 1 4 および 1 2 1 6 、または任意のその他のサブスケジュール ) を生成するために提供される。D b s S u b S c h e d u l e B a s e T e m p l a t e クラス 1 6 0 2 f は、サブスケジュールの基準テンプレート ( 例えば、図 1 9 のサブスケジュール 7 1 0 および 7 1 2 を生成するための基準テンプレート、または図 2 4 のサブスケジュール 1 2 1 4 および 1 2 1 6 を生成するための基準テンプレート ) を取得および / または生成するために提供される。サブスケジュールの基準テンプレートは、サブスケジュールを生成するために必要とされる、基本的な又は基礎的な枠組みを提供する。いくつかの例示的实施では、サブスケジュールのための基準テンプレートは、例えば、ワークステーション 1 0 2 内のサブスケジュール基準テンプレートデータベース ( 図示せず ) から取得されてもよい。

#### 【 0 1 2 4 】

D b s C o m p e l D a t a S e q u e n c e クラス 1 6 0 2 g は、スケジュールに関連付けられた強制データシーケンスを構成するために提供されてもよい。強制データシーケンスは、フィールド装置 ( 例えば、図 1 3 のフィールド装置 1 1 2 a ~ c のうちの 1 つ、または図 2 2 のフィールド装置 1 0 1 2 a ~ c のうちの 1 つ ) が、そのデータ ( 例えば、測定情報、ステータス情報など ) をその他のフィールド装置にいつ通信または発行 ( p

10

20

30

40

50

ublish)すべきかを指定する。図示されている例では、Db s C o m p e l D a t a S e q u e n c e クラス 1 6 0 2 g は、スケジュール(例えば、図 1 9 および図 2 4 のスケジュール 7 1 0 および 1 2 0 2)内で指示されたフィールド装置による機能ブロックの実行に続いて、それらのフィールド装置に、コントローラ(例えば、コントローラ 1 0 6)によって強制データコマンドが発行(issue)されることを確実にする。

#### 【 0 1 2 5 】

D b t S c h e d u l e B a s e T e m p l a t e クラス 1 6 0 2 h および D b t S c h e d u l e クラス 1 6 0 2 i は、一時的な作業領域であって、スケジュール生成プロセスの間に、かつ、フィールド装置にスケジュールを発行(publish)する前に、その中でスケジュール(例えば、スケジュール 7 0 2 または 1 2 0 2、あるいは任意のその他のスケジュール)を生成するための一時的な作業領域を提供する。

10

#### 【 0 1 2 6 】

図 2 9 は、本明細書に記載された例示的な方法に従ってスケジュールを生成するために使用され得る例示的な装置 1 7 0 0 の詳細なブロック図である。例示的な装置 1 7 0 0 は、ワークステーション 1 0 2 (図 1 3)、コントローラ 1 0 6 (図 1 3)、あるいは、ハードウェア、ファームウェア、および/またはソフトウェアの任意の所望の組み合わせを使用して実施されてもよい。例えば、1 つ以上の集積回路、個別半導体部品、または受動電子部品が使用されてもよい。それに加えて、または別法として、例示的な装置 1 7 0 0 のブロックのいくつかまたはすべて、あるいはそれらの部分は、例えばプロセッサシステム(例えば、図 3 5 の例示的なプロセッサシステム 2 3 1 0)によって実行された場合に、図 3 0 ~ 3 3 のフローチャートで表される動作を実行する、機械アクセス可能媒体上に記憶された、命令、コード、ならびに/あるいはその他のソフトウェアおよび/またはファームウェアなどを使用して実施されてもよい。ソフトウェアの例示的实施では、以下で述べる例示的な装置のブロックは、図 2 8 に関連して上述したオブジェクト指向プログラミングクラス 1 6 0 2 a ~ g を実施するために使用されてもよい。

20

#### 【 0 1 2 7 】

例示的な装置 1 7 0 0 は、フィールド装置(例えば、図 1 3 のフィールド装置 1 1 2 a ~ c、または図 2 2 のフィールド装置 1 0 1 2 a ~ c)と情報を交換するように構成されたフィールド装置インタフェース 1 7 0 2 を備える。例えば、フィールド装置インタフェース 1 7 0 2 は、フィールド装置 1 1 2 a ~ c または 1 0 1 2 a ~ c のそれぞれに関連付けられたブロック走査レートまたはブロック未加工実行周期( $T_{BRE}$ )を取得してもよい。

30

#### 【 0 1 2 8 】

例示的な装置 1 7 0 0 は、1 つ以上のフィールド装置(例えば、図 1 3 のフィールド装置 1 1 2 a ~ c、または図 2 2 のフィールド装置 1 0 1 2 a ~ c)であって、それらについてのスケジュール(例えば、図 1 9 のスケジュール 7 0 2、または図 2 4 のスケジュール 1 2 0 2)が生成されるべき 1 つ以上のフィールド装置を含むモジュールと情報を交換するように構成された、モジュールインタフェース 1 7 0 4 をさらに備える。例えば、モジュールインタフェース 1 7 0 4 は、GUI 制御システム設計インタフェースを使用して指定される機能ブロックの相互接続(例えば、図 1 4 に示す機能ブロック 1 2 2 a ~ e の相互接続)によって定義されるモジュール(例えば、図 1 3 のモジュール 1 2 0)から、機能ブロック(例えば、図 1 3 および図 1 4 の機能ブロック 1 2 2 a ~ e)の実行順序を取得するように構成されてもよい。

40

#### 【 0 1 2 9 】

例示的な装置 1 7 0 0 は、スケジュールの基準テンプレート(例えば、図 1 9 のスケジュール 7 0 2 を生成するための基準テンプレート、または図 2 4 のスケジュール 1 2 0 2 を生成するための基準テンプレート)を取得および/または生成するように構成された、スケジュール基準テンプレートインタフェース 1 7 0 6 をさらに備える。例示的实施では、スケジュール基準テンプレートインタフェース 1 7 0 6 は、例えば、モジュール(例えば、図 1 3 のモジュール 1 2 0、または図 2 2 のモジュール 1 0 2 0)内の機能ブロック

50

(例えば、図13の機能ブロック122a~e、および図22の1022a~e)の数量およびタイプに基づいて、ワークステーション102内のスケジュール基準テンプレートデータベースからスケジュールの基準テンプレートを取得する。

#### 【0130】

例示的な装置1700は、本明細書に記載されたスケジュール(例えば、図19のスケジュール702、図24のスケジュール1202、または任意のその他のスケジュール)を生成するように構成されたスケジュール生成器(schedule generator)1708をさらに備える。それに加えて、例示的な装置1700は、本明細書に記載されたサブスケジュール(例えば、図19のサブスケジュール710および712、図24のサブスケジュール1214および1216、または任意のその他のサブスケジュール)を生成するように構成されたサブスケジュール生成器(sub-schedule generator)1710を備える。サブスケジュールの基準テンプレートを生成または取得するために、例示的な装置1700はサブスケジュール基準テンプレートインタフェース1712を備える。

10

#### 【0131】

例示的な装置1700は、スケジュールに関連付けられた強制データシーケンスを生成するように構成された強制データシーケンス生成器(compel data sequence generator)1714をさらに備える。例えば、スケジュール生成器1708がスケジュール(例えば、スケジュール702(図19)または1202(図24)のうちの1つ)の生成を完了した場合、強制データシーケンス生成器1714は、スケジュール内で指示されたフィールド装置による機能ブロックの実行に続いて、それらのフィールド装置にコントローラ(例えば、図13のコントローラ106)が強制データコマンドを発行(issue)することを確実にするために、スケジュールのための強制データシーケンスを生成してもよい。このようにして、フィールド装置によって生成されたデータ(例えば、測定情報、ステータス情報、計算結果など)は、他の動作を実行するためにそのデータを必要とするその他のフィールド装置に通信または発行(publish)される。

20

#### 【0132】

例示的な装置1700は、機能ブロック(例えば、図13および図14の機能ブロック122a~e、ならびに図24の機能ブロック1022a~e)を追加、削除、および相互接続することによってプロセス制御ルーチンを設計するために使用されるGUI設計ソフトウェアアプリケーション(例えば、図14に示すように機能ブロック122a~eを相互接続するために使用される、GUI設計ソフトウェアアプリケーション)から、情報、コマンドなどを受信するように構成された構成インタフェース1716をさらに備える。例えば、ユーザが、2つの機能ブロックの間の相互接続を変更した場合、GUI設計ソフトウェアアプリケーションは、構成インタフェース1716に、変更の通知と変更について説明する情報(例えば、機能ブロックAの出力が機能ブロックBの入力に接続された)とを送信する。さらに、構成インタフェース1716は、プロセス制御システムの構成情報(例えば、デジタルデータバスの結合規則、デジタルデータバスの制限など)を取得するために、ワークステーション102内の、または図13のLAN108に結合された任意のその他のプロセッサシステム内のデータ構造またはデータベースにアクセスするように構成される。

30

40

#### 【0133】

例示的な装置1700は、上記の式1に従ってブロック未加工実行周期( $T_{BRE}$ )を決定するための、未加工実行周期決定器(raw execution period determiner)1718をさらに備える。例示的な装置1700は、図20の丸めテーブル800に示された丸め値に従ってブロック未加工実行周期( $T_{BRE}$ )を丸めるための丸め器(rounder)1720をさらに備える。さらに、例示的な装置1700は、機能ブロックについての開始時間オフセットの数量( $Q_S$ )と開始時間オフセット値( $t_S$ )とを生成するように構成された開始時間オフセット決定器(start tim

50

e offset determiner) 1722を備える。例えば、開始時間オフセット決定器1722は、式3および式4に関連して上述した計算を実行するように構成されてもよい。さらに、例示的な装置1700は、機能ブロック(例えば、図13および図14の機能ブロック122a~e、ならびに図22の機能ブロック1022a~e)のブロック走査レート因子( $F_{BSR}$ )を決定するための、ブロック走査レート因子決定器(block scan rate factor determiner)1724を備える。例えば、ブロック走査レート因子決定器1724は、上記の式2に基づいて、ブロック走査レート因子( $F_{BSR}$ )を決定してもよい。値を比較するために、例示的な装置1700は比較器(comparator)1726を備える。例えば、比較器1726は、フィールド装置に関連付けられた、最も遅いブロック未加工実行周期( $T_{SBE}$ )を、その装置に割り当てられた機能ブロックのブロック未加工実行周期( $T_{BRE}$ )を比較することによって、そして、どれが最も遅いかを決定することによって決定するために使用されてもよい。

10

#### 【0134】

図30~図33は、図29の例示的な装置1700を実施するために使用されてもよい例示的な方法のフローチャートである。いくつかの例示的な実施では、図30~図33の例示的な方法は、プロセッサ(例えば、図35の例示的なプロセッサシステム2310内に示されているプロセッサ2312)による実行のためのプログラムを含む、機械読み取り可能命令を使用して実施されてもよい。プログラムは、CD-ROM、フロッピー(登録商標)ディスク、ハードドライブ、デジタル多用途ディスク(DVD)、またはプロセッサ2312に関連付けられたメモリなどの、有形の媒体上に記憶されたソフトウェア内に組み込まれてもよく、かつ/または、ファームウェアおよび/または専用ハードウェア内に、周知の方法で組み込まれてもよい。さらに、例示的なプログラムは、図30~図33に示すフローチャートに関して説明されるが、図29の例示的な装置1700を実施する多くのその他の方法が代わりに使用されてもよいということを当業者は容易に理解するであろう。例えば、ブロックの実行順序は変更されてもよく、かつ/または、記載されるブロックのうちのいくつかは変更または除去されてもよく、あるいは組み合わせられてもよい。

20

#### 【0135】

説明の目的のために、図30~図33のフロー図については、図24のスケジュール1202に関連して以下で述べる。しかし、図30~図33のフロー図に関連して述べる例示的な方法は、例えば、図19のスケジュール702を含む任意のその他のスケジュールを生成するために使用されてもよい。

30

#### 【0136】

図30を参照すると、スケジュール(図19および図24のスケジュール702および1202、または、プロセス制御システム内の機能ブロックの実行をスケジュールするための任意のその他のスケジュール)を生成するための例示的な方法において、構成インタフェース1716は、新しいモジュール(例えば、図22のモジュール1020)が作成されたか、または既存のモジュールの構成が変更されたかどうかを判定する(ブロック1802)。新しいモジュールが追加されておらず、かつ、既存のモジュールが変更されていないと構成インタフェース1716が判定した場合、新しいモジュールが追加されたか、または、既存のモジュールが変更されたら構成インタフェース1716が判定するまで、制御はブロック1802において留まる。

40

#### 【0137】

新しいモジュールが追加されたか、または、既存のモジュールの構成が変更されたら構成インタフェース1716が判定した場合(ブロック1802)、構成インタフェース1716は、公知の方法に従ってスケジュールを生成すべきかどうかを判定する(ブロック1804)。例えば、構成インタフェース1716は、ワークステーション102(図13)内に記憶されたデータ構造から、スケジュールタイプ記述子を取得してもよい。公知の方法に従って生成された例示的なスケジュールは図18に関連して上述した。スケジュー

50

ールを生成するために公知の方法が使用されるべきであると構成インタフェース 1716 が判定した場合、図 30 の例示的な方法は終了する。

【0138】

スケジュールを生成するために公知の方法が使用されるべきではない、と構成インタフェース 1716 が判定した場合、構成インタフェース 1716 は、モジュール構成がいかなる結合規則にも違反していないということを確認する(ブロック 1806)。この結合規則は、特定のモジュール構成が有効かどうかを示す。例えば、図 22 のモジュール 1020 内の機能ブロック 1022 a ~ e のうちの 1 つが、フィールド装置 1012 a ~ c のうちの 1 つにも、その他のいかなるフィールド装置にも割り当てられていない場合、または、機能ブロック 1022 a ~ e のうちの 1 つが適切に接続されていない場合、モジュール 1020 はスケジュール生成のために有効ではない。図示されている例では、構成インタフェース 1716 は、モジュール構成を結合規則と照合するプロセス制御システム設計ソフトウェアアプリケーション内の確認関数から確認を受信する。モジュール構成が有効ではないと構成インタフェース 1716 が判定した場合(ブロック 1808)、図 30 の例示的な方法は終了する。

10

【0139】

モジュール構成が有効であると構成インタフェース 1716 が判定した場合(ブロック 1808)、図 22 の機能ブロック 1022 a ~ e の間の機能ブロック接続は、デジタルデータバス 1014 (図 22) に、相互に結合される(ブロック 1810)。例えば、機能ブロック接続は、図 15 に関連して上述したように結合されてもよい。例示的な装置 1700 は、次に、機能ブロック 1022 a ~ e (図 22) のそれぞれについて、開始時間オフセットの数量 ( $Q_s$ ) を決定する(ブロック 1812)。ブロック 1812 の動作は、図 31 に関連して以下で述べる例示的な方法を使用して実施されてもよい。

20

【0140】

例示的な装置 1700 が、開始時間オフセットの数量 ( $Q_s$ ) を決定した後で、構成インタフェース 1716 は、デジタルデータバス 1014 の制限を取得する(ブロック 1814)。例えば、構成インタフェース 1716 は、デジタルデータバス 1014 上で動作してもよい最長の可能なスケジュールを定義する時間値を取得してもよい。構成インタフェース 1716 は、例えば、ワークステーション 102 またはコントローラ 106 内のデジタルデータバス特性データベースから制限を取得してもよい。スケジュール基準テンプレートインタフェース 1706 (図 29) は、次に、例えば、ワークステーション 102 内のスケジュールテンプレートデータベースから、スケジュール基準テンプレートを取得する(ブロック 1816)。例示的な装置 1700 は、次に、図 32 の例示的な方法に関連して以下で詳細に述べるように、スケジュール 1202 (図 24) を生成する(ブロック 1818)。例示的な装置 1700 がスケジュール 1202 を生成した後で、制御は呼び出し元の関数またはプロセスに戻り、図 30 の例示的な方法は終了する。

30

【0141】

図 31 を参照すると、図示されているフロー図は、機能ブロック 1022 a ~ e (図 22) のそれぞれについて開始時間オフセットの数量 ( $Q_s$ ) を決定するための図 30 のブロック 1812 を実施するために使用され得る例示的な方法を示す。最初に、モジュールインタフェース 1704 は、図 22 のモジュール 1020 のモジュール実行周期 ( $T_{ME}$ ) を取得する(ブロック 1902)。モジュールインタフェース 1704 は、次に、モジュール 1020 に関連付けられたフィールド装置(例えば、フィールド装置 1022 a ~ e のうちの 1 つ)を選択する(ブロック 1904)。例えば、モジュールインタフェース 1704 は、機能ブロック 1022 a ~ e がどのフィールド装置(例えば、フィールド装置 1012 a ~ c)に割り当てられているかを判定し、そして、例えばフィールド装置オブジェクト(例えば、図 26 のフィールド装置オブジェクト 1402 a ~ c のうちの 1 つ)を選択することによって、それらのフィールド装置のうちの 1 つを選択してもよい。説明の目的のために、ブロック 1904 において、モジュールインタフェース 1704 は図 22 のフィールド装置 1012 a を選択するものとする。オブジェクト指向プログラミング

40

50

環境では、モジュールインタフェース 1704 は、図 26 のフィールド装置オブジェクト 1402a を選択することによってフィールド装置 1012a を選択する。

【0142】

モジュールインタフェース 1704 は、次に、選択されたフィールド装置 1012a に割り当てられた機能ブロック（例えば、機能ブロック 1022a、1022c、または 1022e のうちの 1 つ）を選択する（ブロック 1906）。説明の目的のために、モジュールインタフェース 1704 は、ブロック 1906 において図 22 の AI12 機能ブロック 1022c を選択するものとする。オブジェクト指向プログラミング環境では、モジュールインタフェース 1704 は、図 26 の FFAI12 機能ブロックオブジェクト 1404b を選択することによって AI12 機能ブロック 1022c を選択する。ブロック走査レート因子決定器 1724（図 29）は、次に、選択された AI12 機能ブロック 1022c のブロック走査レート因子（ $F_{BSR}$ ）を決定する（ブロック 1908）。例えば、ブロック走査レート因子決定器 1724 は、モジュール 1020 のモジュール実行周期（ $T_{ME}$ ）と、AI12 機能ブロック 1022c のブロック走査レート（BSR）とを、モジュールインタフェース 1704 から取得し、そして、上記の式 2 を使用して AI12 機能ブロック 1022c のブロック走査レート因子（ $F_{BSR}$ ）を決定してもよい。

【0143】

未加工実行周期決定器 1718（図 29）は、次に、AI12 機能ブロック 1022c のブロック未加工実行周期（ $T_{BRE}$ ）を決定する（ブロック 1910）。例えば、未加工実行周期決定器 1718 は、上記の式 1 を使用して、AI12 機能ブロック 1022c のブロック未加工実行周期（ $T_{BRE}$ ）を、モジュール 1020 のモジュール実行周期（ $T_{ME}$ ）と、AI12 機能ブロック 1022c のブロック走査レート因子（ $F_{BSR}$ ）とに基づいて決定してもよい。丸め器 1720（図 29）は、次に、AI12 機能ブロック 1022c のブロック未加工実行周期（ $T_{BRE}$ ）を、図 20 の丸めテーブル 800 に基づいて、丸められたブロック未加工実行周期（ $RT_{BRE}$ ）に丸める（ブロック 1912）。

【0144】

モジュールインタフェース 1704 は、次に、選択されたフィールド装置 1012a に割り当てられた別の機能ブロックをモジュール 1020 が含むかどうかを判定する（ブロック 1914）。選択されたフィールド装置 1012a に割り当てられた別の機能ブロックをモジュール 1020 が含む、とモジュールインタフェース 1704 が判定した場合（ブロック 1914）、制御はブロック 1906 に戻され、そして、モジュールインタフェース 1704 は、フィールド装置 1012a に割り当てられた次の機能ブロック（例えば、機能ブロック 1022a および 1022e のうちの 1 つ）を選択する。そうではなく、選択されたフィールド装置 1012a に割り当てられた別の機能ブロックをモジュール 1020 が含まないとモジュールインタフェース 1704 が判定した場合（例えば、フィールド装置 1012a に割り当てられた、モジュール 1020 内の機能ブロック 1022a、1022c、および 1022e のすべてについて、丸められたブロック未加工実行周期（ $RT_{BRE}$ ）が決定済みである場合）、比較器 1726 は、次に、選択されたフィールド装置に関連付けられた最も遅いブロック未加工実行周期（ $T_{SBE}$ ）を決定する（ブロック 1916）。例えば、比較器 1726 は、フィールド装置 1012a に割り当てられた機能ブロック 1022a、1022c、および 1022e の、ブロック 1906、1908、1910、および 1912 に関連して上記で決定された丸められたブロック未加工実行周期（ $RT_{BRE}$ ）を比較してもよい。比較器 1726 は、次に、比較に基づいて、最も遅いブロック未加工実行周期（ $T_{SBE}$ ）を丸められたブロック未加工実行周期（ $RT_{BRE}$ ）のうちの最も遅い 1 つに等しく設定してもよい。

【0145】

モジュールインタフェース 1704 は、次に、機能ブロックであって、それについての、開始時間オフセットの数量（ $Q_S$ ）を決定すべき機能ブロックを選択する（ブロック 1918）。例えば、モジュールインタフェース 1704 は、モジュール 1020 内に含まれ、かつ、選択されたフィールド装置 1012a に割り当てられている機能ブロック 10

10

20

30

40

50

22a、1022c、および1022eのうちの1つを選択する。説明の目的のために、モジュール1020は、ブロック1918においてAI12機能ブロック1022cを選択するものとする。開始時間オフセット決定器1722(図29)は、次に、AI12機能ブロック1022cについての開始時間オフセットの数量( $Q_s$ )を決定する(ブロック1920)。例えば、開始時間オフセット決定器1722は、上記の式3を使用して、開始時間オフセットの数量( $Q_s$ )を、AI12機能ブロック1022cの丸められたブロック未加工実行周期( $RT_{BRE}$ )と、ブロック1916において決定された最も遅いブロック未加工実行周期( $T_{SBE}$ )とに基づいて決定してもよい。

【0146】

モジュールインタフェース1704は、次に、選択されたフィールド装置1012aに割り当てられた別の機能ブロックをモジュール1020が含むかどうかを判定する(ブロック1922)。選択されたフィールド装置1012aに割り当てられた別の機能ブロックをモジュール1020が含む、とモジュールインタフェース1704が判定した場合(ブロック1922)、制御はブロック1918に戻され、そして、モジュールインタフェース1704は、フィールド装置1012aに割り当てられた次の機能ブロック(例えば、機能ブロック1022aおよび1022eのうちの1つ)を選択する。そうではなく、選択されたフィールド装置1012aに割り当てられた別の機能ブロックをモジュール1020が含まないとモジュールインタフェース1704が判定した場合(例えば、フィールド装置1012aに割り当てられた、モジュール1020内の機能ブロック1022a、1022c、および1022eのすべてについて、開始時間オフセットの数量( $Q_s$ )が決定済みである場合)、モジュールインタフェース1704は、別のフィールド装置(例えば、図22のフィールド装置1012bおよび1012cのうちの1つ)がモジュール1020に関連付けられているかどうかを判定する(ブロック1924)。例えば、別のフィールド装置(例えば、フィールド装置1012bおよび1012c)に割り当てられている任意の機能ブロック(例えば、機能ブロック1022bおよび1022c)であって、それについての、開始時間オフセットの数量( $Q_s$ )がまだ決定されていない機能ブロックをモジュール1020が含む場合、モジュールインタフェース1704は、別のフィールド装置がモジュール1020に関連付けられていると判定してもよい。

【0147】

別のフィールド装置(例えば、図22のフィールド装置1012bおよび1012cのうちの1つ)がモジュール1020に関連付けられている、とモジュールインタフェース1704が判定した場合(ブロック1924)、制御はブロック1904に戻され、その時点で、モジュールインタフェース1704は次のフィールド装置(例えば、図22のフィールド装置1012bおよび1012cのうちの1つ)を選択する。そうではなく、別のフィールド装置がモジュール1020に関連付けられていないとモジュールインタフェース1704が判定した場合(ブロック1924)、モジュールインタフェース1704は、別のモジュール(例えば、モジュール1020以外のモジュール)がデジタルデータバス1014に関連付けられているかどうかを判定する(ブロック1926)。例えば、デジタルデータバス1014に関連付けられた別のモジュールは、フィールド装置1012a~cのうちのいくつかに割り当てられた機能ブロックを含んでもよく、そして、例示的な装置1700は、次に、それらの機能ブロックについての開始時間オフセットの数量( $Q_s$ )を決定する。別のモジュールがデジタルデータバス1014に関連付けられているとモジュールインタフェース1704が判定した場合、制御はブロック1902に戻され、その時点で、モジュールインタフェース1704は次のモジュールのモジュール実行周期( $T_{ME}$ )を取得する。そうではなく、別のモジュールがデジタルデータバス1014に関連付けられていないとモジュールインタフェース1704が判定した場合、制御は呼び出し元の関数またはプロセス(例えば、図30の例示的な方法)に戻され、図31の例示的な方法は終了する。

【0148】

図32Aおよび図32Bを参照すると、図示されているフロー図は、サブスケジュール

10

20

30

40

50

(例えば、図24のサブスケジュール1214および1216)、ならびにサブスケジュールに基づくスケジュール(例えば、図24のスケジュール1202)を生成するための、図30のブロック1818を実施するために使用され得る例示的な方法を示す。最初に、サブスケジュール生成器1710は、デジタルデータバス1014に関連付けられた、すべての丸められたブロック未加工実行周期( $RT_{BRE}$ )を取得する。例えば、サブスケジュール生成器1710は、デジタルデータバス1014に関連付けられたすべての機能ブロック(例えば、図22の機能ブロック1022a~e)について、ブロック1912において決定された、すべての丸められたブロック未加工実行周期( $RT_{BRE}$ )を取得する。サブスケジュール基準テンプレートインタフェース1712(図29)は、次に、丸められたブロック未加工実行周期( $RT_{BRE}$ )のそれぞれについてのサブスケジュール基準テンプレートを取得する(ブロック2004)。例えば、サブスケジュール基準テンプレートインタフェース1712は、ワークステーション102(図13)内のサブスケジュール基準テンプレートデータベースまたはデータ構造から、サブスケジュール基準テンプレートを取得してもよい。図示されている例では、サブスケジュール基準テンプレートのそれぞれは、ブロック2002において取得された丸められたブロック未加工実行周期( $RT_{BRE}$ )のうちのそれぞれの1つに等しいサブスケジュール実行周期を有するように構成されている。例えば、図24に示すように、サブスケジュール1214のサブスケジュール実行周期は500msに等しく、サブスケジュール1216のサブスケジュール実行周期は2000msに等しい。

#### 【0149】

サブスケジュール生成器1710は、最も短いサブスケジュール実行周期を有するサブスケジュールテンプレートを選択する(ブロック2006)。例えば、サブスケジュール生成器1710は、比較器1726(図29)を使用して、すべてのサブスケジュール実行周期を相互に比較してサブスケジュール周期のうちどの1つが最も短いかを判定してもよい。サブスケジュール生成器1710は、次に、同期データ転送を実行するように構成されるフィールド装置を選択する(ブロック2008)。フィールド装置(例えば、図22のフィールド装置1012a~cのうちの一つ)は、同期した方法でデータを転送する機能ブロック(例えば、図22の機能ブロック1022a~eのうちの一つ)を実行するように割り当てられている場合、同期データ転送を実行するように構成される。機能ブロックは、その機能ブロックによって生成されたデータが別の機能ブロックによって特定の時に必要とされる場合、同期した方法でデータを転送する。例えば、PID13機能ブロック1022d(図22および図24)が、AI12機能ブロック1022c(図22および図24)からのデータを図24に示す時間 $t_2$ までに必要とする場合、フィールド装置1012aは、PID13機能ブロック1022dが時間 $t_2$ までにデータを取得することを確実にするために、AI12機能ブロック1022cに関連付けられた同期データ転送を実行しなければならない。説明の目的のために、サブスケジュール生成器1710はフィールド装置1012a(図22および図24)を選択するものとする。

#### 【0150】

サブスケジュール生成器1710は、次に、選択されたフィールド装置1012aに関連付けられた、最も遅いブロック未加工実行周期( $T_{SBE}$ )を取得する(ブロック2010)。例えば、サブスケジュール生成器1710は、フィールド装置1012aについて図31のブロック1916において決定された、最も遅いブロック未加工実行周期( $T_{SBE}$ )を取得してもよい。サブスケジュール生成器1710は、次に、例えば比較器1726(図29)を使用して、最も遅いブロック未加工実行周期( $T_{SBE}$ )が選択されたサブスケジュールテンプレートのサブスケジュール実行周期に等しいかどうかを判定する(ブロック2012)。ブロック2006において選択されたサブスケジュールテンプレートを使用して生成されるサブスケジュールは、選択されたサブスケジュールテンプレートのサブスケジュール実行周期に等しい最も遅いブロック未加工実行周期( $T_{SBE}$ )に関連付けられたフィールド装置によって実行されるように割り当てられた機能ブロックのみについての開始時間オフセット( $t_s$ )を含んでもよい。したがって、ブロック2008にお

10

20

30

40

50

いて選択されたフィールド装置 1012a の最も遅いブロック未加工実行周期 ( $T_{SBE}$ ) が、選択されたサブスケジュールテンプレートのサブスケジュール実行周期に等しくないとサブスケジュール生成器 1710 が判定した場合 (ブロック 2012)、そのフィールド装置はスキップされ、そして、制御はブロック 2008 に戻され、その時点でサブスケジュール生成器 1710 は同期データ転送を実行するように構成される別のフィールド装置を選択する。

#### 【0151】

そうではなく、ブロック 2008 において選択されたフィールド装置 1012a の最も遅いブロック未加工実行周期 ( $T_{SBE}$ ) が選択されたサブスケジュールテンプレートのサブスケジュール実行周期に等しいとサブスケジュール生成器 1710 が判定した場合 (ブロック 2012)、サブスケジュール生成器 1710 は、選択されたフィールド装置 1012a に割り当てられ、かつ、同期データ転送に関連付けられた機能ブロック (例えば、機能ブロック 1022a、1022c、または 1022e のうちの 1 つ) を選択する (ブロック 2014)。説明の目的のために、サブスケジュール生成器 1710 は AI12 機能ブロック 1022c (図 22 および図 24) を選択するものとする。

#### 【0152】

サブスケジュール生成器 1710 は、次に、選択された AI12 機能ブロック 1022c のデータ利用可能時間 ( $t_{DA}$ ) を取得する (ブロック 2016)。図示されている例では、すべてのデータ利用可能時間 ( $t_{DA}$ ) は、同期データ利用可能時間生成器 (synchronous data available time generator) (図示せず) によってあらかじめ定められ、機能ブロックの入力ポートに関連したデータ構造内に記憶される。例えば、同期データ利用可能時間生成器は、図 30 のスケジュール生成プロセスに先立って、またはそのプロセスの間に、すべての機能ブロック (例えば、機能ブロック 1022a ~ e) を走査して、どの機能ブロックが同期した方法でそれらに転送されるデータを必要とするかを判定してもよい。同期データ利用可能時間生成器は、次に、他の機能ブロックに同期した方法でデータを転送しなければならない機能ブロックが、実行のための、およびデジタルデータバス 1014 を介してそれらのデータを転送するための十分な時間を有することを確実にして、特定の時にデータが必要とする機能ブロックがその時までデータを受信することができるようにするためのデータ利用可能時間 ( $t_{DA}$ ) をあらかじめ定めてもよい。別の機能ブロックからのデータを必要とすることなしに実行されてもよい機能ブロック (例えば、AI11 機能ブロック AI11a) については、データ利用可能時間 ( $t_{DA}$ ) は 0 に設定され、なぜなら、その機能ブロックはサブスケジュール周期が始まったらすぐに実行されてもよいからである。

#### 【0153】

開始時間オフセット決定器 1722 (図 29) は、図 33 に関連して以下に述べるように、選択された AI12 機能ブロック 1022c についての開始時間オフセット値 ( $t_s$ ) を決定する (ブロック 2018)。サブスケジュール生成器 1710 は、次に、ブロック 2018 において決定された開始時間オフセット値 ( $t_s$ ) のためのデータ転送時間を割り当てる (ブロック 2020)。データ転送時間は、1 つの機能ブロックから別の機能ブロックに、デジタルデータバス 1014 を介してデータを通信するために必要とされる時間である。同じ装置内の別の機能ブロックにデータを転送する機能ブロックは、データ転送時間を有さない。しかし、別の装置内の別の機能ブロックにデータを転送する機能ブロックは、データ転送時間を有する。データ転送時間は、デジタルデータバス 1014 上の時間を占め、したがって、すべての機能ブロックの実行とデジタルデータバス 1014 上でのデータ転送とが行われることが可能であることを確実にするのに十分なだけサブスケジュールが長いことを保証するように、サブスケジュールを生成するために必要とされる。

#### 【0154】

サブスケジュール生成器 1710 は、次に、フィールド装置 1012a に割り当てられた別の機能ブロックが、同期データ転送に関連付けられているかどうかを判定する (プロ

10

20

30

40

50

ック2022)。フィールド装置1012aに割り当てられた別の機能ブロックが、同期データ転送に関連付けられている、とサブスケジュール生成器1710が判定した場合(ブロック2022)、制御はブロック2014に戻され、その時点で、サブスケジュール生成器1710は同期データ転送に関連付けられた次の機能ブロックを選択する。それ以外の場合、サブスケジュール生成器1710は、デジタルデータバス1014に通信可能に結合された別のフィールド装置が、同期データ転送を実行するように構成されるかどうかを判定する(ブロック2024)。デジタルデータバス1014に通信可能に結合された別のフィールド装置が同期データ転送を実行するように構成されるとサブスケジュール生成器1710が判定した場合(ブロック2024)、制御はブロック2008に戻され、その時点で、サブスケジュール生成器1710は同期データ転送を実行するように構成される別のフィールド装置(例えば、図22および図24のフィールド装置1012bおよび1012cのうちの1つ)を選択する。

10

**【0155】**

そうではなく、デジタルデータバス1014に通信可能に結合された別のフィールド装置は同期データ転送を実行するように構成されないとサブスケジュール生成器1710が判定した場合(ブロック2024)、サブスケジュール生成器1710は非同期データ転送を処理する(ブロック2026)。例えば、サブスケジュール生成器1710は、非同期データ転送に関連付けられた機能ブロック1022a~eのうちのいくつかを識別し、それらの機能ブロックについてのデータ利用可能時間( $t_{DA}$ )を、ブロック2016に関連して上述した方法に類似した方法で取得し、開始時間オフセット値( $t_s$ )を、図33

20

**【0156】**

サブスケジュール生成器1710は、次に、ブロック2006において選択されたサブスケジュールテンプレートのサブスケジュール実行周期が、上述のように決定された開始時間オフセット値( $t_s$ )とデータ転送時間とを含むのに十分な長さであるかどうかを判定する(ブロック2028)。ブロック2006において選択されたサブスケジュールテンプレートのサブスケジュール実行周期が、開始時間オフセット値( $t_s$ )とデータ転送時間とを含むのに十分な長さではないとサブスケジュール生成器1710が判定した場合(ブロック2028)、サブスケジュール生成器1710は、サブスケジュールを再生成するために、開始時間オフセット値( $t_s$ )とデータ転送時間とを破棄する。サブスケジュールテンプレートのために生成される機能ブロックの開始時間オフセット値( $t_s$ )とデータ転送時間とを含むのに十分な長さのサブスケジュール実行周期を、ブロック2006において選択されたサブスケジュールテンプレートに提供するために、サブスケジュール生成器1710は、サブスケジュールテンプレートのサブスケジュール周期を拡張する(ブロック2032)。すなわち、サブスケジュール生成器1710は、サブスケジュールテンプレートのサブスケジュール実行周期に時間を追加して、サブスケジュール実行周期を長くする。例えば、図24の2000msのサブスケジュール1216を生成するために使用されるサブスケジュールテンプレートが、機能ブロック1022a~cおよび1022eに関連付けられた、開始時間オフセット( $t_s$ )とデータ転送時間とを含むのに十分な長さのサブスケジュール実行周期を有さない場合、サブスケジュール生成器1710は、サブスケジュールテンプレートに500ms(または、任意のその他の時間)を追加してもよい。

30

40

**【0157】**

サブスケジュール生成器1710が、選択されたサブスケジュールテンプレートのサブスケジュール実行周期を拡張した後で(ブロック2032)、制御はブロック2008に戻され、その時点で、ブロック2008、2010、2012、2014、2016、2018、2020、2022、2024、2026、および2028の動作が、拡張されたサブスケジュール実行周期に基づいて繰り返される。ブロック2006において選択さ

50

れたサブスケジュールテンプレートのサブスケジュール実行周期が、開始時間オフセット値 ( $t_s$ ) とデータ転送時間とを含むのに十分な長さであるとサブスケジュール生成器 1710 が判定した場合 (ブロック 2028)、サブスケジュール生成器 1710 は、別のサブスケジュールを生成すべきかどうかを判定する (ブロック 2034)。例えば、図 24 のスケジュール 1202 を生成するために図 30 の例示的な方法が使用され、かつ、サブスケジュール生成器 1710 がサブスケジュール 1214 を生成済みであり、しかしサブスケジュール 1216 は生成済みではない場合、サブスケジュール生成器 1710 は、別のサブスケジュール (例えば、サブスケジュール 1216) が生成されるべきであると判定する。別のサブスケジュール (例えば、サブスケジュール 1216) が生成されるべきであるとサブスケジュール生成器 1710 が判定した場合 (ブロック 2034)、制御はブロック 2006 (図 32A) に戻され、その時点で、サブスケジュール基準テンプレートインタフェース 1712 は、次に最も短いサブスケジュール実行周期 (例えば、図 24 の 2000ms のサブスケジュールに対応する 2000ms のサブスケジュール実行周期) を選択する (ブロック 2006)。

#### 【0158】

それ以外の場合、スケジュール生成器 1708 は、スケジュール (例えば、図 24 のスケジュール 1202) を生成するために、サブスケジュール生成器 1710 によって生成されたサブスケジュール (例えば、図 24 のサブスケジュール 1214 および 1216) をマージする (ブロック 2036)。強制データシーケンス生成器 1714 は、次に、スケジュールのための強制データシーケンスを生成する (ブロック 2038)。強制データシーケンスは、フィールド装置 1012a ~ c が、それらのそれぞれの機能ブロック 1022a ~ e のうちのいくつかに対応するデータを、生成されたスケジュールに従ってデジタルデータバス 1014 上に発行 (publish) することをもたらすために、コントローラ 106 がフィールド装置 1012a ~ c に強制データコマンドを通信することを確実にする。制御は、次に、呼び出し元の関数またはプロセス (例えば、図 30 の例示的な方法) に戻され、図 32A および図 32B の例示的な方法は終了する。

#### 【0159】

図 33 を参照すると、図示されているフロー図は、機能ブロックについての開始時間オフセット値 ( $t_s$ ) を決定するための、図 32A のブロック 2018 を実施するために使用されてもよい例示的な方法を示す。図示されている例では、図 33 の例示的な方法は、開始時間オフセット値 ( $t_s$ ) を決定するために、上記の式 4 を使用する。最初に、開始時間オフセット決定器 1722 (図 29) は、選択された機能ブロック (例えば、図 32A のブロック 2014 において選択された機能ブロック) に関連付けられた、開始時間オフセットの数量 ( $Q_s$ ) を取得する (ブロック 2102)。説明の目的のために、AI12 機能ブロック 1022c (図 22 および図 24) が、選択された機能ブロックであるとする。したがって、ブロック 2102 において取得される、開始時間オフセットの数量 ( $Q_s$ ) は 4 である。図示されている例では、選択された AI12 機能ブロック 1022c についての、開始時間オフセットの数量 ( $Q_s$ ) (例えば、4) は、図 30 のブロック 1812 において決定される。開始時間オフセット決定器 1722 は、次に、開始時間オフセットカウンタを 0 にリセットする (ブロック 2104)。開始時間オフセットカウンタの値は、上記の式 4 の順序番号 ( $N_{seq}$ ) を提供するために使用される。さらに、開始時間オフセットカウンタの値は、選択された機能ブロックについての開始時間オフセット値 ( $t_s$ ) がすべて決定された場合を判定するために使用される。

#### 【0160】

開始時間オフセット決定器 1722 は、第 1 の開始時間オフセット値 ( $t_s$ ) を決定する (ブロック 2106)。例えば、開始時間オフセット決定器 1722 は、図 24 において  $t_1$  として示されている AI12 機能ブロック 1022c の開始時間オフセット値 ( $t_s$ ) を、上記の式 4 を使用して、開始時間オフセットカウンタの値 (例えば、現在の開始時間オフセットの順序番号 ( $N_{seq}$ ) (順序 0、順序 1、順序 2 など)) と、AI12 機能ブロック 1022c を実行するために割り当てられたフィールド装置 (例えば、フィール

10

20

30

40

50

ド装置 1012a) に関連付けられた最も遅いブロック未加工実行周期 ( $T_{SBE}$ ) と、AI12機能ブロック 1022c) についてブロック 2016) において取得されたデータ利用可能時間 ( $t_{DA}$ ) とに基づいて決定する。

#### 【0161】

開始時間オフセット決定器 1722) は、次に、開始時間オフセットカウンタの値をインクリメントし (ブロック 2108) 、そして、開始時間オフセットカウンタがブロック 2102) において取得された開始時間オフセットの数量 ( $Q_S$ ) と等しいかどうかを判定する (ブロック 2110) 。開始時間オフセットカウンタは、開始時間オフセットの数量 ( $Q_S$ ) と等しくないと開始時間オフセット決定器 1722) が判定した場合、制御はブロック 2106) に戻り、その時点で、開始時間オフセット決定器 1722) は、次の開始時間オフセット値 ( $t_s$ ) (例えば、図 24) において  $t_s$  として示されている、AI12機能ブロック 1022c) の開始時間オフセット値 ( $t_s$ ) ) を決定する。それ以外の場合、制御は、呼び出し元の関数またはプロセス (例えば、図 32A および図 32B の例示的な方法) に戻され図 33) の例示的な方法は終了する。

#### 【0162】

図 34) は、図 24) のスケジュール 1202) の、例示的な機能実行およびデータ転送シーケンス図 2200) である。例示的な機能実行およびデータ転送シーケンス図 2200) は、機能ブロック 1022a) ~ e) の実行と、1つの機能ブロックから別の機能ブロックにデジタルデータバス 1014) を介してデータを転送するために必要とされるデータ転送時間との間の関係を示す。例示的な機能実行およびデータ転送シーケンス図は、さらに、図 24) のフィールド装置 1012a) ~ c) がデジタルデータバス 1014) 上にデータを発行 (publish) ) することをもたらすために、強制データコマンドがコントローラ 106) によっていつ発行されるかも示す。例えば、図 34) に示すように、AI11機能ブロック実行 2202) の後で PID12) 実行 2208) を可能にするために、強制データコマンド 2204) が、フィールド装置 1012a) に AI11) データ転送 2206) を実行させる。その他の機能ブロック実行、強制データコマンド、および対応するデータ転送も示されている。図示されていないが、2つの機能ブロックが同じフィールド装置によって実行される場合、機能ブロックの間でデータを交換するために、デジタルデータバス 1014) に関連付けられたデータ転送時間 (例えば、AI11) データ転送 2206) ) は必要ではなく、なぜなら、データ転送は同じフィールド装置内で発生するからである。

#### 【0163】

図 35) は、本明細書に記載された装置および方法を実施するために使用されてもよい、例示的なプロセッサシステム 2310) のブロック図である。図 35) に示すように、プロセッサシステム 2310) は、相互接続バス 2314) に結合されたプロセッサ 2312) を含む。プロセッサ 2312) は、レジスタセットまたはレジスタ空間 2316) を含み、これは、図 35) では、全体的にオンチップであるとして示されているが、代わりに、全体的または部分的にチップ外に配置されて、プロセッサ 2312) に専用の電氣的接続を介して直接結合されてもよく、かつ/または、相互接続バス 2314) を介して結合されてもよい。プロセッサ 2312) は、任意の好適なプロセッサ、処理ユニット、またはマイクロプロセッサであってもよい。図 35) には示されていないが、システム 2310) は、マルチプロセッサシステムであってもよく、したがって、プロセッサ 2312) と同一であるかまたはそれに類似し、かつ、相互接続バス 2314) に通信可能に結合された 1つ以上の追加のプロセッサを含んでもよい。

#### 【0164】

図 35) のプロセッサ 2312) は、メモリコントローラ 2320) と周辺入出力 (I/O) コントローラ 2322) とを含むチップセット 2318) に結合される。周知のように、チップセットは、一般に、I/O およびメモリ管理機能を提供し、さらに、チップセット 2318) に結合された 1つ以上のプロセッサがアクセス可能な、またはそれらのプロセッサによって使用される、複数の汎用および/または専用のレジスタ、タイマなども提供する。メモリコントローラ 2320) は、プロセッサ 2312) (または、複数のプロセッサが存在

10

20

30

40

50

する場合は、それらのプロセッサ)が、システムメモリ2324および大容量記憶メモリ2325にアクセスすることを可能にする機能を実行する。

【0165】

システムメモリ2324は、例えば、スタティックランダムアクセスメモリ(SRAM)、ダイナミックランダムアクセスメモリ(DRAM)、フラッシュメモリ、読み出し専用メモリ(ROM)などの、任意の所望のタイプの揮発性および/または不揮発性メモリを含んでもよい。大容量記憶メモリ2325は、ハードディスクドライブ、光学式ドライブ、テープ記憶装置などを含む、任意の所望のタイプの大容量記憶装置を含んでもよい。

【0166】

周辺I/Oコントローラ2322は、プロセッサ2312が、周辺入出力(I/O)装置2326および2328、ならびにネットワークインタフェース2330と、周辺I/Oバス2332を介して通信することを可能にする機能を実行する。I/O装置2326および2328は、例えば、キーボード、ビデオディスプレイまたはモニタ、マウスなどの任意の所望のタイプのI/O装置であってもよい。ネットワークインタフェース2330は、プロセッサシステム2310が別のプロセッサシステムと通信することを可能にする、例えば、イーサネット(登録商標)装置、非同期転送モード(ATM)装置、802.11装置、DSLモデム、ケーブルモデム、セルラーモデムなどであってもよい。

【0167】

メモリコントローラ2320およびI/Oコントローラ2322は、チップセット2318内の別個の機能ブロックとして図35に示されているが、これらのブロックによって実行される機能は、1つの半導体回路内に統合されてもよく、または2つ以上の別個の集積回路を使用して実施されてもよい。

【0168】

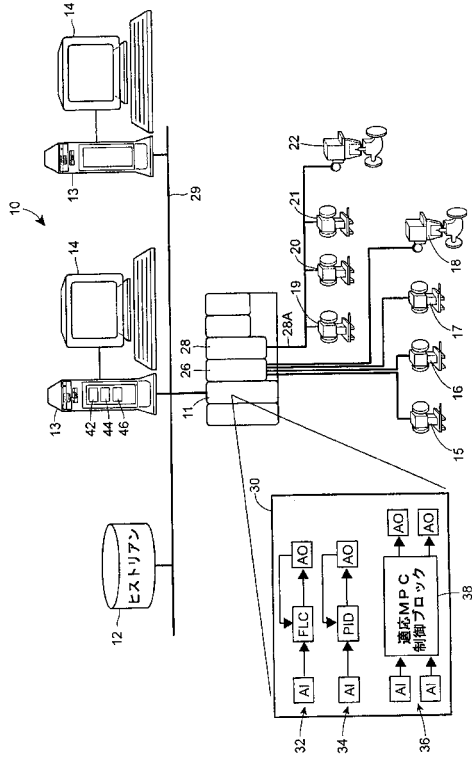
特定の方法、装置、および製品について、本明細書で説明したが、本発明が包含する範囲はそれらに限定されない。反対に、本発明は、字義的に、または均等論に基づいて、添付の特許請求の範囲内に公正に含まれるすべての方法、装置、および製品を包含する。さらに、本明細書では、自動ブロック割り当ておよびスケジューリングルーチンは、ファウンデーション(FOUNDATION(登録商標))フィールドバス(Fieldbus)ネットワークに適用可能であるとして説明したが、これらの技術は、ファウンデーション(FOUNDATION(登録商標))フィールドバス(Fieldbus)ネットワークに限定されず、そして、その代わりに、任意のその他の適用可能な、バスをベースにした通信ネットワークにおいて使用されてもよい。

10

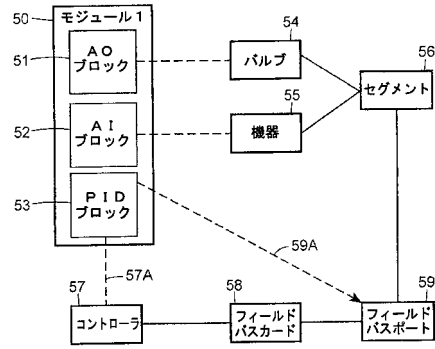
20

30

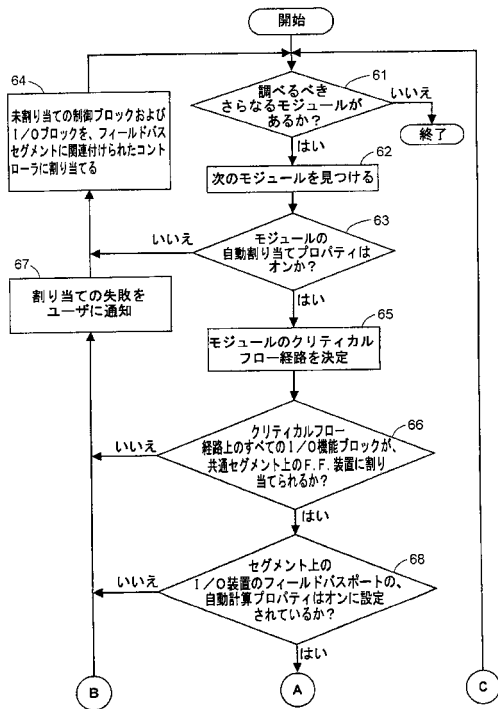
【図1】



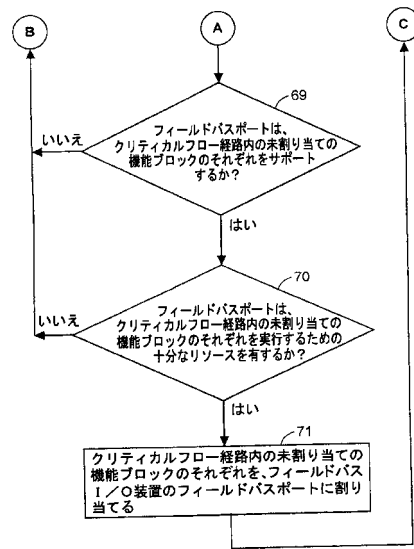
【図2】



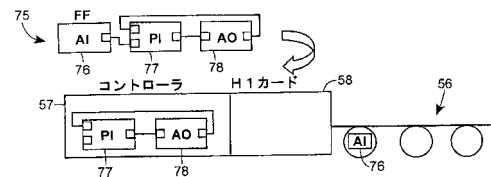
【図3A】



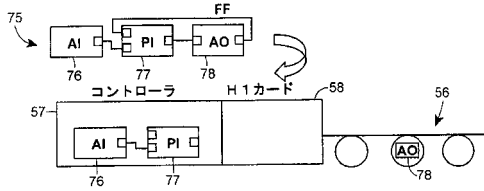
【図3B】



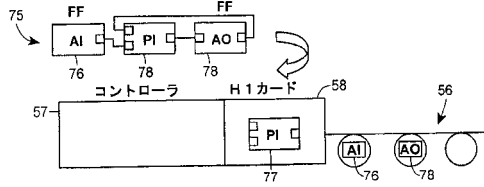
【図4】



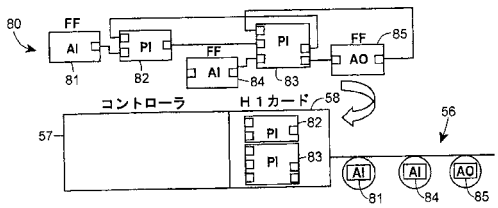
【図5】



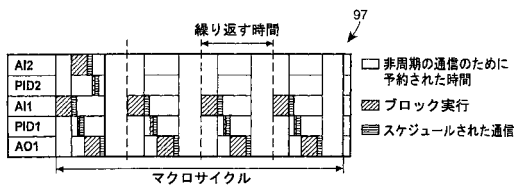
【図6】



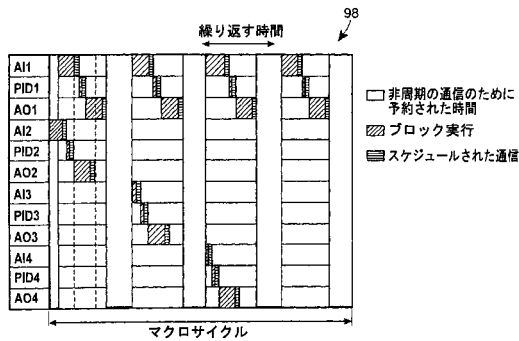
【図7】



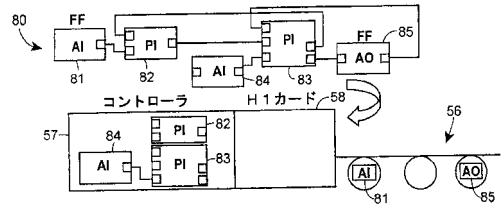
【図11】



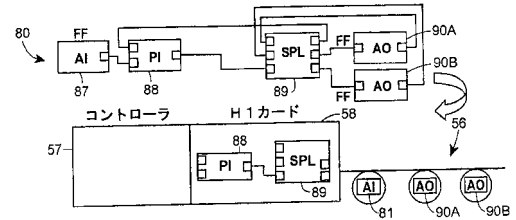
【図12】



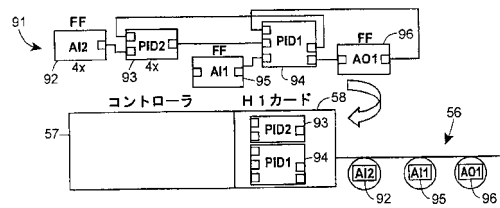
【図8】



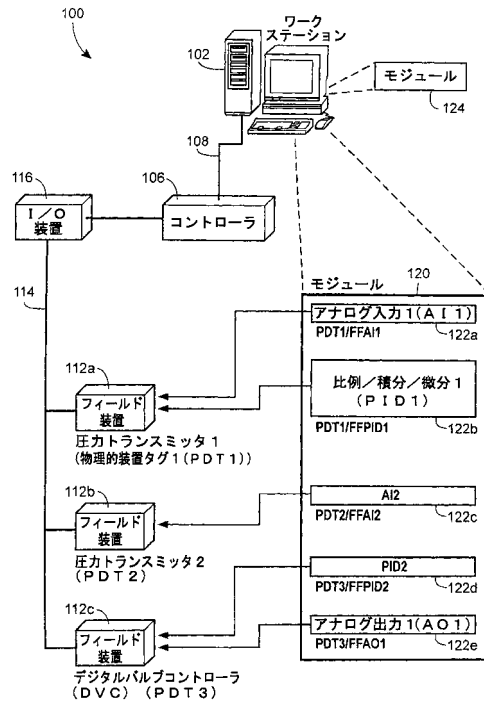
【図9】



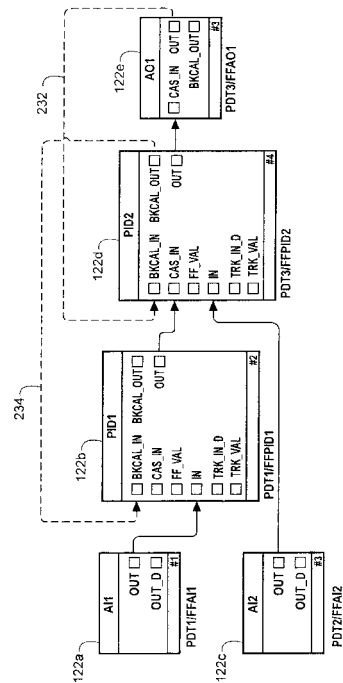
【図10】



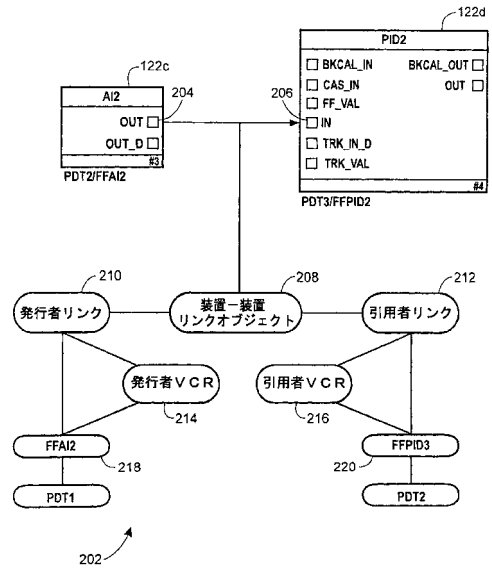
【図13】



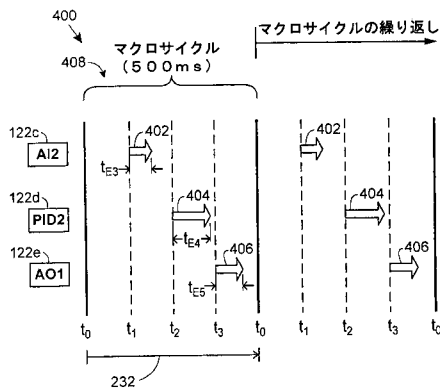
【 図 1 4 】



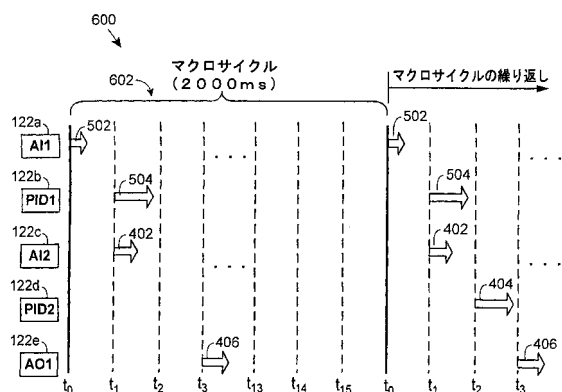
【 図 1 5 】



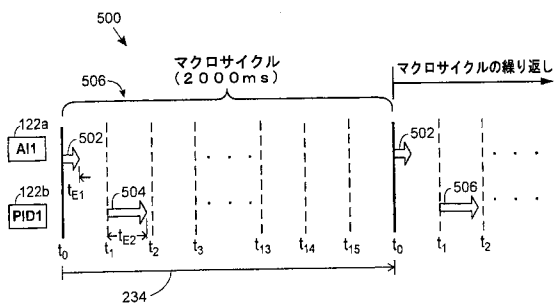
【 図 1 6 】



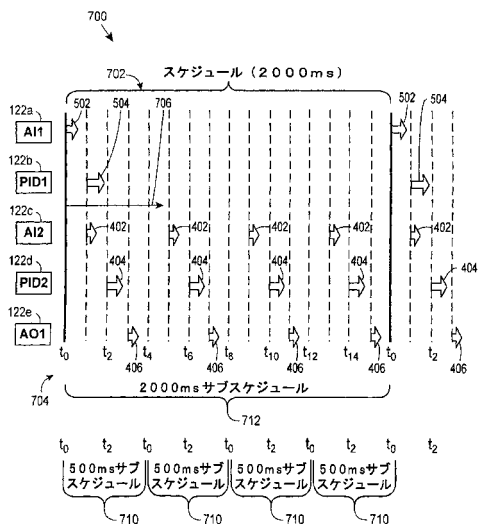
【 図 1 8 】



【 図 1 7 】



【図19】



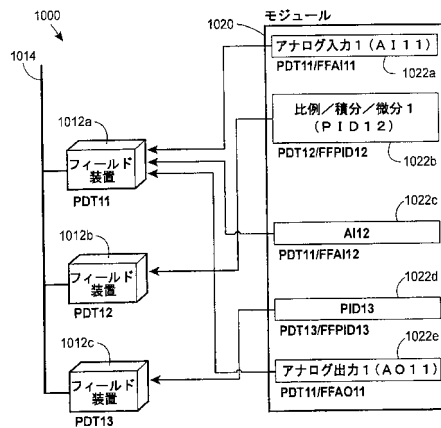
【図20】

ブロック 未加工実行周期 ( $T_{BRE}$ ) (ms)	0~500	>500 ~1000	>1000 ~2000	>2000
丸められたブロック 未加工実行周期 ( $R_{T_{BRE}}$ ) (ms)	500	1000	2000	4000

【図21】

フィールド装置/ 機能ブロック	ブロック 走査レート因子 $F_{BSR}$	$T_{BRE}$ (ms)	$R_{T_{BRE}}$ (ms)	開始時間 オフセットの 数量
PDT1/FFAI1	4	$500 \times 4 = 2000$	2000	1
PDT1/FFPID1	4	$500 \times 4 = 2000$	2000	1
PDT2/FFAI2	1	$500 \times 1 = 500$	500	1
PDT3/FFPID2	1	$500 \times 1 = 500$	500	1
PDT3/FFAO1	1	$500 \times 1 = 500$	500	1

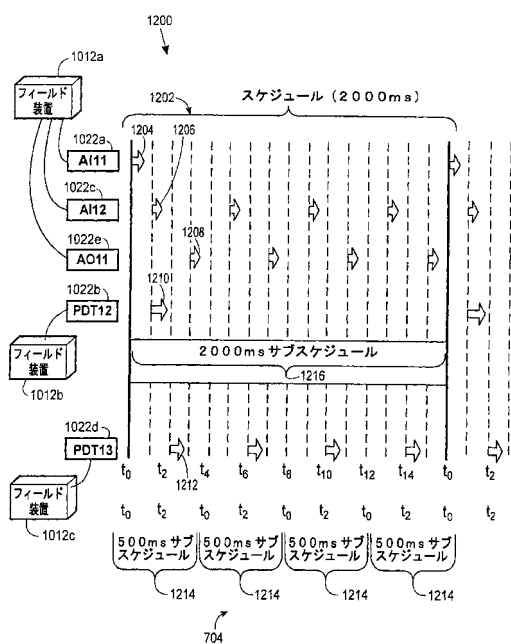
【図22】



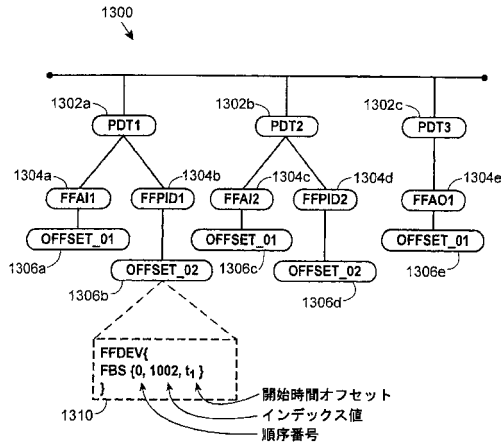
【図23】

フィールド装置/ 機能ブロック	ブロック 走査レート因子 $F_{BSR}$	$T_{BRE}$ (ms)	$R_{T_{BRE}}$ (ms)	開始時間 オフセットの数量
PDT11/FFAI1	4	$500 \times 4 = 2000$	2000	$2000/2000=1$
PDT12/FFPID12	4	$500 \times 4 = 2000$	2000	$2000/2000=1$
PDT11/FFAI2	1	$500 \times 1 = 500$	500	$2000/500=4$
PDT13/FFPID13	1	$500 \times 1 = 500$	500	$500/500=1$
PDT11/FFAO1	1	$500 \times 1 = 500$	500	$2000/500=4$

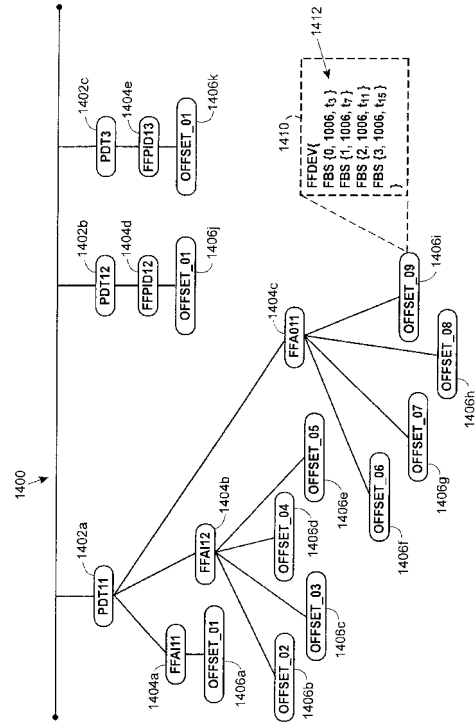
【図24】



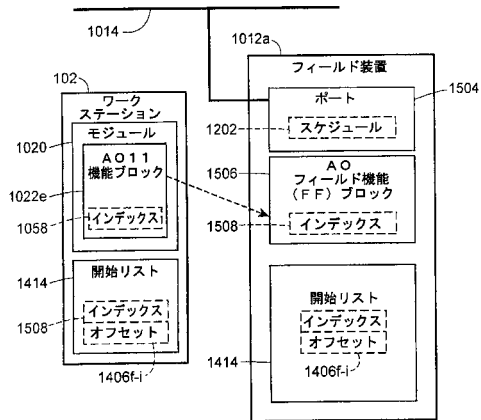
【図 25】



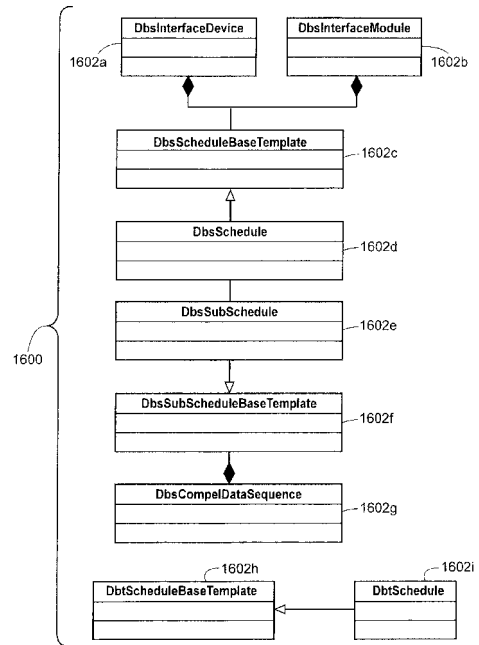
【図 26】



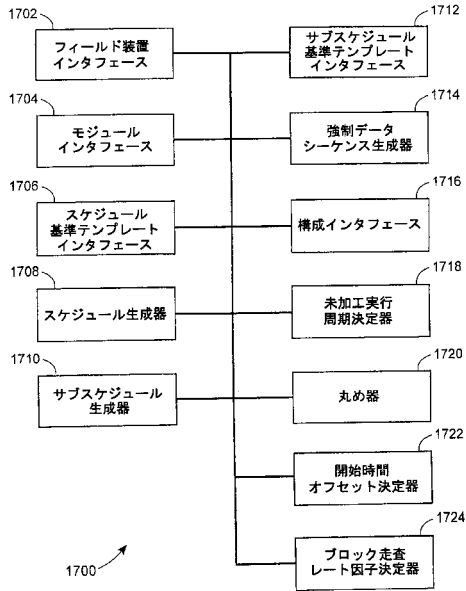
【図 27】



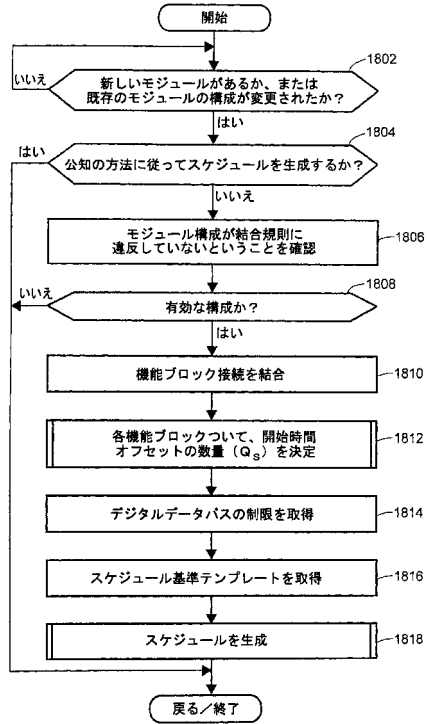
【図 28】



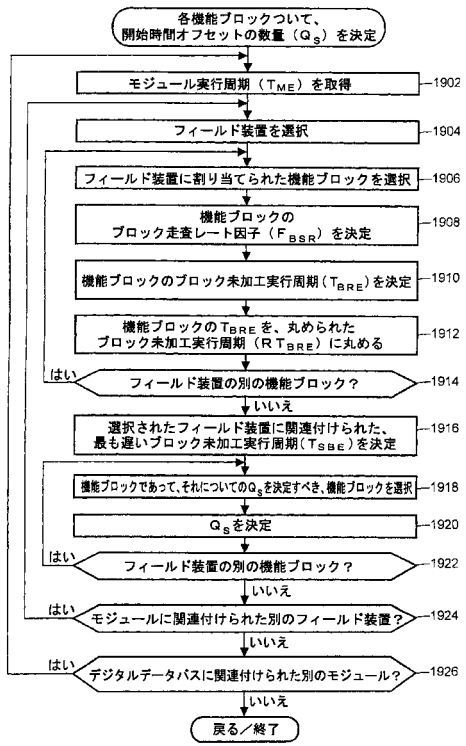
【図 29】



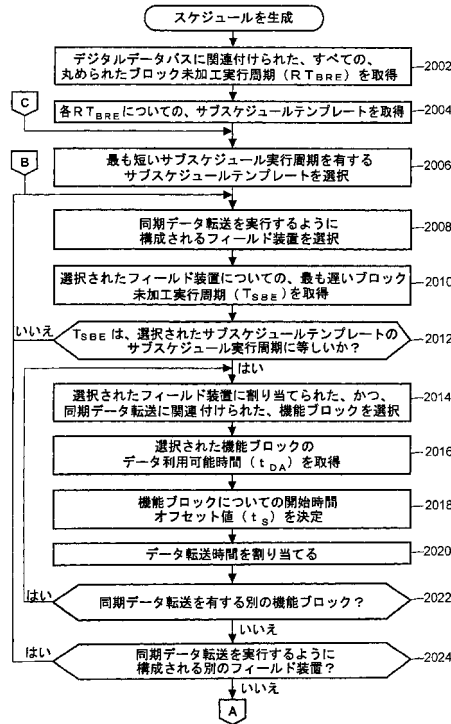
【図 30】



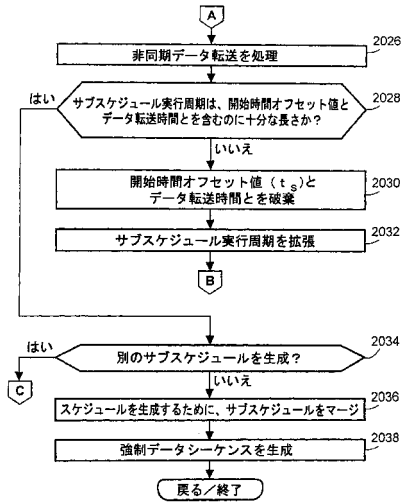
【図 31】



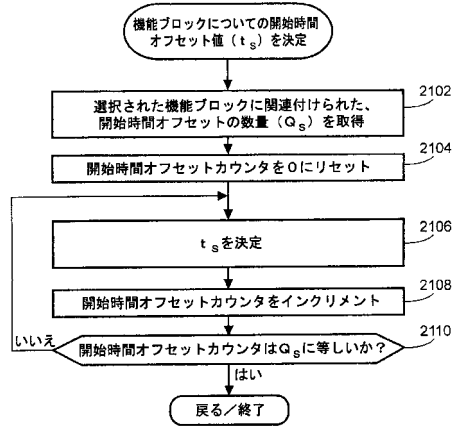
【図 32 A】



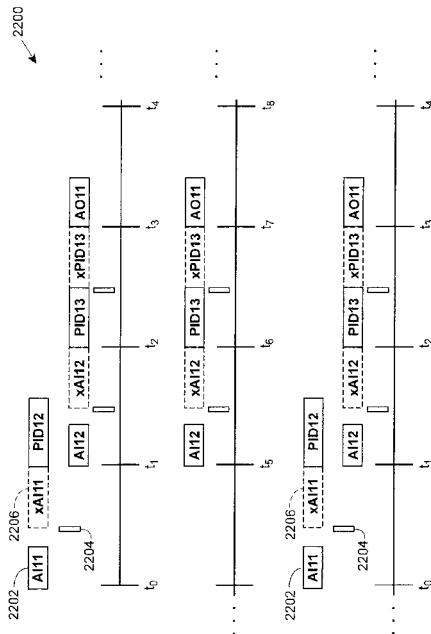
【図32B】



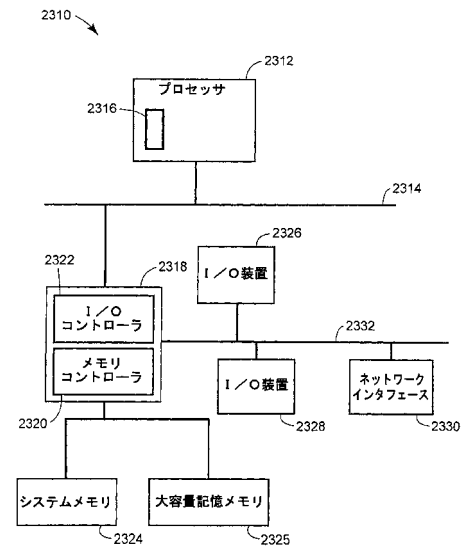
【図33】



【図34】



【図35】



## フロントページの続き

- (72)発明者 アルパー ティー . エンバー  
アメリカ合衆国 テキサス州 78727 オースティン アラメダ トレイス サークル 12  
340
- (72)発明者 ケネス ディー . クリボシェイン  
アメリカ合衆国 テキサス州 78759 オースティン バレシト ドライブ 6700
- (72)発明者 ダニエル ディーン クリステンセン  
アメリカ合衆国 テキサス州 78717 オースティン マーサズ ドライブ 9001
- (72)発明者 ラム ラマチャンドラン  
アメリカ合衆国 テキサス州 78750 オースティン パトリス ドライブ 9801
- (72)発明者 ジョン マイケル ルーカス  
イギリス国 レスターシャー エルイー17 5イーエス レイア ルターワース ウェールズ  
オーチャード 7
- (72)発明者 エプティサム エス . タニユース  
アメリカ合衆国 テキサス州 78613 シーダー パーク ウッド クリーク ドライブ 1  
005

審査官 青山 純

- (56)参考文献 特開2004-265411(JP,A)  
特表2001-501794(JP,A)  
特開2006-048184(JP,A)  
特開2002-163003(JP,A)  
特開2002-055713(JP,A)  
特開平10-177401(JP,A)

- (58)調査した分野(Int.Cl., DB名)  
G05B 19/05  
H04L 12/28